

**PENERAPAN SISTEM PAKAR MENDIAGNOSIS PENYAKIT
MATA RABUN DENGAN METODE *FORWARD*
CHAINING BERBASIS *MOBILE***

SKRIPSI

**HAFIZ AL IRSYAD
71154049**



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2020**

**PENERAPAN SISTEM PAKAR MENDIAGNOSIS PENYAKIT
MATA RABUN DENGAN METODE *FORWARD*
CHAINING BERBASIS *MOBILE***

SKRIPSI

Diajukan untuk memenuhi syarat mencapai Gelar Sarjana Komputer

**HAFIZ AL IRSYAD
71154049**



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA
MEDAN
2020**

PERSETUJUAN SKRIPSI

Hal : Surat Persetujuan Skripsi

Lamp : -

Kepada Yth:

Dekan Fakultas Sains dan Teknologi

Universitas Islam Negeri Sumatera Utara Medan

Assalamu'alaikum Wr, Wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengatakan perbaikan, maka kami selaku pembimbing berpendapat bahwa skripsi saudara :

Nama : Hafiz Al Irsyad

Nomor Induk Mahasiswa : 71154049

Program Studi : Ilmu Komputer

Judul : Penerapan Sistem Pakar Mendiagnosis Penyakit
Mata Rabun Dengan Metode Forward Chaining
Berbasis Mobile.

Dapat disetujui untuk segera di *munaqosyahkan*. Atas perhatiannya kami ucapkan terima kasih.

Medan, 13 Februari 2020 M

19 Rajab 1442 H

Komisi Pembimbing,

Dosen Pembimbing I

Dosen Pembimbing II

Dr. Mhd. Furqan, S.Si., M.Comp.Sc.

Yusuf Ramadhan Nasution, M.Kom.

NIP. 198008062006041003

NIB. 1100000075

SURAT PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan dibawah ini:

Nama : Hafiz Al Irsyad
Nomor Induk Mahasiswa : 71154049
Program Studi : IlmuKomputer
Judul : Penerapan Sistem Pakar Mendiagnosis Penyakit
Mata Rabun Dengan Metode Forward Chaining
Berbasis Mobile.

Menyatakan bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya. Apabila di kemudian hari ditemukan plagiat dalam skripsi ini maka saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi lainnya sesuai dengan peraturan yang berlaku.

Medan,13 Februari 2020

Hafiz Al Irsyad
NIM.71154049



KEMENTERIAN AGAMA REPUBLIK INDONESIA
UNIVERSITAS ISLAM NEGERI SUMATERA UTARA MEDAN
FAKULTAS SAINS DAN TEKNOLOGI
Jl.IAIN No.1 Medan KodePos 20335
Telp.(061)6615683-6622925, Fax. (061)6615683
Url:<http://saintek.uinsu.ac.id>, E-mail: saintek@uinsu.ac.id

PENGESAHAN SKRIPSI

Nomor.005/ST/ST.V.2/PP.01.1/01/2021

Judul : Penerapan Sistem Pakar Mendiagnosis Penyakit
Mata Rabun Dengan Metode Forward Chaining
Berbasis Mobile.

Nama : Hafiz Al Irsyad
Nomor Induk Mahasiswa : 71154049
Program Studi : Ilmu Komputer
Fakultas : Sains dan Teknologi

Telah dipertahankan dihadapan Dewan Penguji Skripsi Program Studi Ilmu
Komputer Fakultas Sains dan Teknologi UIN Sumatera Utara Medan dan
dinyatakan **LULUS**.

Pada hari/tanggal : 13 Februari 2020
Tempat : Ruang Sidang Fakultas Sains dan Teknologi

Tim Ujian Munaqasyah
Ketua,

Dr. Mhd. Furqan, S.Si., M.Comp.Sc
NIP. 198008062006041003

Dewan Penguji,

Penguji I

Penguji II

Dr. Mhd. Furqan, S.Si., M.Comp.Sc.
NIP. 198008062006041003

Yusuf Ramadhan Nasution, M Kom.
NIB. 1100000075

Penguji III

Penguji IV

Sriani, M.Kom.
NIB. 1100000108

Armansyah, M.Kom.
NIB. 1100000074

Mengesahkan,
Dekan Fakultas Sains dan Teknologi
UIN Sumatera Utara Medan

Dr. H. M. Jamil, M.A.
NIP. 196609101999031002

ABSTRAK

Android disesuaikan juga di peradaban manusia. Perkembangan teknologi informasi mempengaruhi banyak bidang seperti pekerjaan, kegiatan diluar rumah, kendaraan dan lain-lain. Kesehatan mata merupakan anugerah yang tidak dapat dibayar. Melihat betapa pentingnya mata bagi kehidupan kita, menjaga kesehatan mata menjadi hal utama yang perlu kita lakukan setiap hari. Menjaga kesehatan mata sangatlah penting karena sayangnya penglihatan kita hilang. Banyak yang berpendapat bahwa di Asia Tenggara banyak orang yang menderita penyakit mata, terutama miopia. Sebelum terjadi sesuatu pada mata, sebaiknya kita merawatnya dengan baik. Jenis rabun dekat saat semua jenis miopia sebenarnya adalah ketidakaturan dalam pemfokusan pada bayangan benda yang terlihat atau kelainan refraksi (*ametropia*). Menggunakan sistem pakar dan menerapkan metode rantai langsung yang dirancang untuk memudahkan pasien dalam mengidentifikasi jenis penyakit mata rabun. Dan sebagai pengetahuan bagi para pengguna internet tentang pentingnya menjaga kesehatan mata dari segi gejala mata miopia pada sistem. Aplikasi ini dapat mendiagnosis dan mengetahui hasil diagnosa penyakit mata rabun pada pasien dengan menggunakan metode rantai langsung yaitu metode rantai langsung, berlaku untuk aplikasi dalam bentuk membangun hubungan aturan dengan memilih gejala dan solusi. Gejala dan solusi dimasukkan satu per satu sesuai dengan gejalanya. Sistem aplikasi *mobile* dapat digunakan pengguna dalam mendiagnosis gejala dan mendapatkan pengetahuan terkait mata rabun, aplikasi ini dapat menemukan solusi untuk jenis penyakit mata rabun sesuai dengan gejala yang dialami pasien.

Kata Kunci : Mata Rabun, Metode Forward Chaining, Berbasis Mobile

ABSTRACT

Android is also adapted to human civilization. The development of information technology affects many fields such as work, activities outside the home, vehicles and others. Eye health is a gift that cannot be paid. Seeing how important our eyes are to our lives, maintaining eye health is the main thing we need to do every day. Maintaining eye health is very important because unfortunately our vision is lost. Many argue that in Southeast Asia many people suffer from eye diseases, especially myopia. Before anything happens to the eye, we should take good care of it. The type of nearsightedness when any type of myopia is actually an irregularity in focusing on the image of the visible object or refractive error (ametropia). Using an expert system and applying a direct chain method designed to make it easier for patients to identify the type of myopic eye disease. And as knowledge for internet users about the importance of maintaining eye health in terms of eye symptoms of myopia in the system. This application can diagnose and find out the results of myopic eye disease diagnosis in patients using the direct chain method, namely the direct chain method, applicable to applications in the form of building relationship rules by selecting symptoms and solutions. Symptoms and solutions are entered one by one according to the symptoms. The mobile application system can be used by users in diagnosing symptoms and getting knowledge related to myopic eye, this application can find solutions for types of myopic eye disease according to the symptoms experienced by the patient.

Keywords: Myopia, Forward Chaining Method, Mobile Based

KATA PENGANTAR



Assalamu'alaikum Warahmatullahi Wabarakatuh.

Alhamdulillahirobbil'alamin. Segala puji dan syukur penulis panjatkan kehadirat Allah SWT, karena dengan izin-Nya lah penulis dapat menyelesaikan Skripsi ini. Shalawat beserta salam semoga senantiasa tercurah kepada junjungan penulis, yakni Nabi Muhammad SAW.

Skripsi ini untuk melengkapi serta memenuhi syarat untuk menjalankan program mata kuliah yang ada di semester VIII Ilmu Komputer Universitas Islam Negeri Sumatera Utara untuk mendapat gelar Strata-1. Dalam penyusunan Skripsi ini penulis menyadari adanya keterbatasan, kemampuan, dan wawasan penulis, maka penulis menerima masukan dan saran dari semua pihak. Untuk itu penulis menyampaikan rasa terima kasih kepada :

1. Prof. Dr. Saidurrahman, M.Ag. selaku Rektor UIN Sumatera Utara Medan.
2. Dr. H. M. Jamil, M.A. selaku Dekan Fakultas Sains dan Teknologi UIN Sumatera Utara Medan.
3. Dr. Mhd. Furqan, S.Si, M.Comp.Sc. Selaku Ketua Program Studi Ilmu Komputer Fakultas Sains dan Teknologi UIN Sumatera Utara Medan, serta dosen-dosen dan staff administrasi yang telah membantu selama proses perkuliahan.
4. Dr. Mhd. Furqan, S.Si, M.Comp.Sc dan Yusuf Ramadhan Nasution M.Kom selaku Pembimbing Skripsi yang telah memberikan motivasi dan bimbingan selama proses penyelesaian skripsi.
5. Rakhmat Kurniawan R, S.T, M.Kom selaku Kepala Laboratorium Fakultas Sains dan Teknologi UIN Sumatera Utara Medan yang telah memfasilitasi penelitian dalam rangka penyelesaian skripsi.
6. Bapak Ilka Zufria, M.Kom selaku Sekretaris Program studi Ilmu Komputer Universitas Islam Negeri Sumatera Utara Medan.

7. Ibu Sriani,M.Kom selaku Penguji I Saya yang selalu mengarahkan dan memberikan saran terhadap skripsi saya.
8. Bapak Armansyah,M.Kom Penguji II Saya yang selalu mengarahkan dan memberikan saran untuk skripsi saya.
9. Bapak Abdul Halim Hasugian,M.Kom Selaku Penasehat Akademik saya yang telah banyak membantu dan memberikan masukan beserta saran dalam bidang komprehensif.
10. Teristimewa buat Ayah dan Mama tercinta, kedua adikku yang terus menerus memberikan semangat, pengertian dan do'a untuk dapat menyelesaikan pendidikan ini dengan sebaik-baiknya yang mempunyai makna tersendiri bagi penulis dalam penyelesaian penulisan Skripsi ini.
11. Buat teman-teman kelas dan yang tidak dapat disebut satu persatu yang selalu mensupport penulis dari awal kuliah hingga akhir.

Dalam penulisan ini banyak kesulitan yang penulis hadapi, namun demikian berkat bimbingan dan bantuan dari berbagai pihak akhirnya penulis dapat menyelesaikan. Demi kelancaran dalam penyelesaian laporan proposal skripsi ini tidak terlepas dari bantuan pihak terutama kepada Ayah dan Ibu yang telah memberikan bantuan moril maupun materil, semangat dan do'a yang begitu besar kepada penulis.

Medan, Februari 2020

Penulis

(Hafiz Al Irsyad)

DAFTAR ISI

ABSTRAK	I
ABSTRACT	II
KATA PENGANTAR.....	III
DAFTAR ISI.....	V
DAFTAR GAMBAR.....	VII
DAFTAR TABEL	IX
DAFTAR LAMPIRAN	X
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
BAB II TINJAUAN PUSTAKA.....	4
2.1 Kecerdasan Buatan (<i>Artificial Intelligent</i>)	4
2.2 Sistem Pakar.....	4
2.3 Metode Runut maju (<i>Forward Chaining</i>).....	6
2.4 Bahasa Pemrograman <i>Java</i>	9
2.5 Aplikasi Berbasis Mobile.....	10
2.6 Mata Rabun	11
2.7 Riset Terkait.....	12
BAB III METODE PENELITIAN	15
3.1 Tempat dan Waktu Penelitian	15
3.1.1 Tempat Penelitian.....	15
3.1.2 Waktu & Jadwal Pelaksanaan Peneliti.....	15
3.2 Bahan dan Alat Penelitian.....	15
3.2.1 Perangkat Keras	15
3.2.2 Perangkat Lunak	15

3.3 Cara Kerja	15
3.3.1 Perencanaan	15
3.3.2 Teknik Pengumpulan Data	16
3.3.3 Analisa Kebutuhan	18
3.3.4 Perancangan	19
3.3.5 Pengujian	20
3.3.6 Penerapan / Penggunaan	21
BAB IV HASIL DAN PEMBAHASAN	22
4.1 Pembahasan	22
4.1.1 Analisis Data	22
4.1.2 Representasi Data	22
4.1.3 Hasil Analisis Data	24
4.1.4 Perancangan Sistem	29
4.2 Hasil	37
4.2.1 Pengujian	37
4.2.2 Penerapan	60
BAB V KESIMPULAN DAN SARAN	62
5.1 Kesimpulan	62
5.2 Saran	62
DAFTAR PUSTAKA	63
LAMPIRAN	65

DAFTAR GAMBAR

Gambar 3.1 Tahap – Tahap Perencanaan.....	16
Gambar 3.2 <i>Forward Chaining</i> Sistem.....	20
Gambar 4.1 Perancangan <i>Diagram</i> Sistem	29
Gambar 4.2 <i>Diagram</i> Form Pendaftaran.....	29
Gambar 4.3 <i>Diagram</i> Main Menu.....	29
Gambar 4.4 <i>Diagram</i> Form Konsultasi.....	29
Gambar 4.5 <i>Diagram</i> Form Solusi.....	30
Gambar 4.6 <i>Diagram</i> Form Hasil Konsultasi	30
Gambar 4.7 Perancangan <i>Flowchart</i> Sistem.....	31
Gambar 4.8 Perancangan <i>Flowchart Form</i> Registrasi.....	32
Gambar 4.9 Perancangan <i>Flowchart Form</i> Main Menu	32
Gambar 4.10 Perancangan <i>Flowchart Form</i> Konsultasi.....	32
Gambar 4.11 Perancangan <i>Flowchart Form</i> Solusi.....	33
Gambar 4.12 Gambar Tampilan <i>Form</i> Registrasi.....	34
Gambar 4.13 Gambar Tampilan <i>Form</i> Main Menu	35
Gambar 4.14 Gambar Tampilan <i>Form</i> Konsultasi.....	36
Gambar 4.15 Gambar Tampilan <i>Form</i> Solusi.....	37
Gambar 4.16 Input Data Registrasi Rabun Jauh	38
Gambar 4.17 Input Data Registrasi Kasus ke 2 Rabun Jauh.....	39
Gambar 4.18 Input Data Konsultasi Rabun Jauh.....	40
Gambar 4.19 Input Data Konsultasi Kasus ke 2 Rabun Jauh	41
Gambar 4.20 Proses <i>Forward Chaining</i> Rabun Jauh.....	42
Gambar 4.21 Proses <i>Forward Chaining</i> Kasus ke 2 Rabun Jauh.....	43
Gambar 4.22 Input Data Registrasi Rabun Dekat.....	44
Gambar 4.23 Input Data Registrasi Kasus ke 2 Rabun Dekat	45
Gambar 4.24 Input Data Konsultasi Rabun Dekat.....	46
Gambar 4.25 Input Data Konsultasi Kasus ke 2 Rabun Dekat	47
Gambar 4.26 Proses <i>Forward Chaining</i> Rabun Dekat	48
Gambar 4.27 Proses <i>Forward Chaining</i> Kasus ke 2 Rabun Dekat.....	49

Gambar 4.28 Input Data Registrasi Katarak Senilis	50
Gambar 4.29 Input Data Registrasi Kasus ke 2 Katarak Senilis.....	51
Gambar 4.30 Input Data Konsultasi Katarak Senilis	52
Gambar 4.31 Input Data Konsultasi Kasus Ke 2 Katarak Senilis.....	53
Gambar 4.32 Proses <i>Forward Chaining</i> Katarak Senilis.....	54
Gambar 4.33 Proses <i>Forward Chaining</i> Kasus ke 2 Katarak Senilis	55

DAFTAR TABEL

Tabel 3.1 Data Gejala.....	16
Tabel 3.2 Data Penyakit	17
Tabel 3.3 Keputusan Penyakit Mata Rabun	18
Tabel 3.4 Keterangan Perancangan Diagram.....	19
Tabel 4.1 Data Gejala.....	22
Tabel 4.2 Data Penyakit	23
Tabel 4.3 Keputusan Penyakit Mata Rabun	24
Tabel 4.4 Basis Data Pengetahuan	25
Tabel 4.5 Atural (<i>Rule</i>)	26

DAFTAR LAMPIRAN

Lampiran 1 : Listing Program

Lampiran 2 : Surat Balasan Penelitian

Lampiran 3 : Kartu Bimbingan Skripsi

Lampiran 4 : Daftar Riwayat Hidup

BAB I PENDAHULUAN

1.1 Latar Belakang

Android disesuaikan juga dengan peradaban manusia, android dapat mempengaruhi perkembangan teknologi informasi dengan mempengaruhi banyak bidang seperti pekerjaan, kegiatan diluar rumah, kendaraan dan lain-lain.

وَأَنفِقُوا فِي سَبِيلِ اللَّهِ وَلَا تُلْقُوا بِأَيْدِيكُمْ إِلَى التَّهْلُكَةِ وَأَحْسِنُوا إِنَّ اللَّهَ يُحِبُّ الْمُحْسِنِينَ

Dan penuhi (hartamu) di jalan Allah, dan jangan jatuh (dirimu sendiri) ke dalam kehancuran dengan tanganmu sendiri, dan berbuat baik. Sungguh, Allah menyukai orang yang berbuat baik. (QS. Al-Baqarah: 195)

يَا أَيُّهَا الَّذِينَ آمَنُوا إِذَا قُمْتُمْ إِلَى الصَّلَاةِ فَاغْسِلُوا وُجُوهَكُمْ وَأَيْدِيَكُمْ إِلَى الْمَرَافِقِ وَامْسَحُوا بِرُءُوسِكُمْ وَأَرْجُلَكُمْ إِلَى الْكَعْبَيْنِ وَإِنْ كُنْتُمْ جُنُبًا فَاطَّهَّرُوا وَإِنْ كُنْتُمْ مَرْضَىٰ أَوْ عَلَىٰ سَفَرٍ أَوْ جَاءَ أَحَدٌ مِّنْكُم مِّنَ الْغَائِطِ أَوْ لَمَسْتُمُ النِّسَاءَ فَلَمْ تَجِدُوا مَاءً فَتَيَمَّمُوا صَعِيدًا طَيِّبًا فَامْسَحُوا بِوُجُوْهِكُمْ وَأَيْدِيكُمْ مِنْهُ مَا يُرِيدُ اللَّهُ لِيَجْعَلَ عَلَيْكُمْ مِنْ حَرَجٍ وَلَكِنْ يُرِيدُ لِيُطَهِّرَكُمْ وَلِيُتِمَّ نِعْمَتَهُ عَلَيْكُمْ لَعَلَّكُمْ تَشْكُرُونَ

Hai orang-orang beriman! Jika akan shalat, basuh wajah dan tangan sampai ke siku, usapkan kepala dan (basuh) kaki sampai ke mata kaki. Jika Anda seorang junub, mandilah. Dan jika Anda sakit atau dalam perjalanan, atau kembali dari toilet (toilet) atau menyentuh seorang wanita, maka jika Anda tidak mendapatkan air, maka tanamlah dengan debu yang baik (suci); bersihkan wajah dan tangan Anda dengan itu (debu). Allah tidak ingin mempersulit hidup Anda, tetapi Dia ingin menyucikan Anda dan melengkapi nikmat-Nya untuk Anda sehingga Anda akan bersyukur. (QS. Al-Maidah : 6).

Jika anda tidak menjaga kebersihan seluruh tubuh, atau hanya sebagian, itu tidak akan menyelamatkan anda dari penyakit. Selain itu kebersihan dan kesehatan mata juga harus diperhatikan. Indra penglihatan sangat penting dan kita harus menjaganya, kita terlibat dalam aktivitas dan melakukan berbagai aktivitas, yang semuanya mengandalkan penglihatan atau mata. Banyak orang yang belum mengetahui cara menjaga kesehatan mata dengan melakukan aktivitas atau

aktivitas yang menyebabkan kerusakan mata. Kita sering menjumpai hal-hal seperti salah satu penyakit mata yaitu *miopia*, *hyperopia*, atau katarak.

Penerapan sistem pakar merupakan bidang penelitian dalam kecerdasan buatan yang telah ada selama beberapa dekade. (William P. Wagner, 2017). Penggunaan metode rantai langsung dalam penelitian ini disebabkan karena program aplikasi yang dibuat dari sistem pakar memerlukan masukan dari pengguna dan pendekatannya. Metode rantai langsung merupakan salah satu metode dalam sistem pakar. (Cut Fiarnia, 2015).

Kesehatan mata merupakan anugerah yang tidak dapat dibayar. Melihat betapa pentingnya mata bagi kehidupan kita, menjaga kesehatan mata menjadi hal utama yang perlu kita lakukan setiap hari. Menjaga kesehatan mata sangatlah penting karena ketika kita tidak bisa menjaga kesehatan itu resiko penglihatan kita akan buram dan bisa jadi secara perlahan akan menghilang. Banyak yang berpendapat bahwa di Asia Tenggara banyak orang yang menderita penyakit mata, terutama *miopia*. Sebelum terjadi sesuatu pada mata, sebaiknya kita merawatnya dengan baik. *Miopia* digolongkan menjadi beberapa jenis yaitu rabun jauh dan rabun dekat, di saat semua jenis *miopia* sebenarnya adalah ketidakaturan dalam pemfokusan pada bayangan benda yang terlihat atau kelainan refraksi (*ametropia*).

Berdasarkan uraian diatas, maka penulis memutuskan untuk mengambil judul ***“Perancangan Sistem Pakar Mendiagnosis Penyakit Mata Rabun Dengan Metode Forward Chaining Berbasis Mobile”***.

1.2 Rumusan Masalah

Berdasarkan hal tersebut di atas dirumuskan masalah pokok yang akan dikaji dalam skripsi ini, yaitu:

1. Bagaimana menerapkan Metode *Forward Chaining* Dalam Mendiagnosis Penyakit Mata Rabun?
2. Bagaimana merancang Aplikasi Sistem Pakar Untuk Mendiagnosis Penyakit Mata Rabun Dengan Metode *Forward Chaining* Berbasis *Mobile*?

1.3 Batasan Masalah

Untuk memperoleh hasil pembahasan yang sesuai dengan tujuan yang dimaksudkan maka ditetapkan batasan-batasan masalah yaitu:

1. Bahasa Pemrograman yang digunakan dalam pembuatan aplikasi Mendiagnosis Penyakit Mata Rabun yaitu Bahasa Pemrograman *Java*.
2. Metode digunakan *Forward Chaining*.
3. Penelitian ini hanya meliputi penyakit mata rabun yaitu Rabun Jauh dan Rabun Dekat.
4. Sistem pakar ini dirancang untuk menentukan jenis penyakit mata rabun berdasarkan gejala ringan yang dirasakan.
5. Aplikasi Android Studio.

1.4 Tujuan Penelitian

Untuk mencapai permasalahan diatas, adapun tujuan penelitian ini sebagai berikut :

1. Terapkan metode *forward chaining* dalam diagnosis penyakit mata rabun.
2. Pengembangan sistem pakar untuk diagnosis penyakit mata rabun menggunakan metode *forward chaining* berbasis *mobile*.

1.5 Manfaat Penelitian

Untuk mencapai uraian permasalahan diatas, adapun manfaat penelitian ini sebagai berikut :

- 1 Memberikan solusi-solusi menyelesaikan masalah tentang kesehatan mata yang diterapkan pada aplikasi menerapkan metode *forward chaining*.
- 2 Memudahkan masyarakat dalam mengenali gejala-gejala dan jenis-jenis penyakit mata rabun, serta memudahkan dalam melakukan diagnosis dini terhadap penyakit mata rabun secara cepat dan tepat.
- 3 Mengetahui gejala – gejala mata rabun dengan beberapa pertanyaan yang terdapat di dalam aplikasi tersebut.

BAB II

TINJAUAN PUSTAKA

Pada penelitian ini yang akan dilakukan untuk mendiagnosis mata rabun dengan menggunakan bahasa pemrograman *java* dan metode *forward chaining* dalam berbentuk aplikasi berbasis *mobile*.

2.1 Kecerdasan Buatan (*Artificial Intelligent*)

Artificial Intelligence (Inggris: Artificial Intelligence atau AI) didefinisikan sebagai kecerdasan yang direpresentasikan oleh objek buatan. Sistem seperti itu biasanya dianggap sebagai komputer. Kecerdasan dibuat dan dicolokkan ke mesin (komputer) untuk melakukan pekerjaan dengan cara yang bisa dilakukan manusia.

2.2 Sistem Pakar

Sistem pakar adalah cabang dari AI (kecerdasan buatan) yang mengembangkan pengetahuan khusus untuk memecahkan suatu masalah dengan seorang pakar manusia. Seorang ahli manusia adalah ahli dalam bidang keahlian tertentu, yang berarti bahwa seorang ahli memiliki masalah yang tidak dapat diselesaikan orang lain secara efektif. (Windah Supartini, 2016). Program sistem pakar sangat berguna untuk pengambilan keputusan karena bidang aplikasinya sangat luas. (Yunia Ervinaeni, 2019). Di bawah ini adalah komponen utama dari sistem pakar, yaitu:

A. *Knowledge Base* (Basis Pengetahuan)

Basis pengetahuan terbaik berisi artikel yang ditulis dengan baik yang terus diperbarui, memiliki mesin pencari (mesin pencari) yang baik, dan format konten dan struktur klasifikasi yang dipikirkan dengan cermat.

B. *Inference Engine* (Motor inferensi)

Bertanggung jawab atas analisis pengetahuan dan kesimpulan berdasarkan basis pengetahuan dan aturan. sebuah mesin inferensi yang berisi fungsi mekanisme berpikir dan pola berpikir sistem yang digunakan oleh para ahli. Metode penalaran (dan inferensi) dapat berupa inferensi langsung inferensi balik

C. User Interface

Interaksi pengguna menciptakan dan memelihara basis pengetahuan. Antara muka pengguna sistem pakar dapat berupa program desktop, web, atau seluler. User Interface juga sangat diperlukan di dalam suatu aplikasi yang dimana agar dapat membuat nyaman suatu *client* menggunakan aplikasi itu.

D. Development Engine

Bagian mesin pengembangan merupakan komponen dari sistem pakar, dimana bagian ini diperuntukkan bagi tenaga ahli yang dapat menambah pengetahuan baru pada sistem. Keduanya disimpan di database, kemudian dijadikan untuk mengambil sebuah keputusan yang nantinya akan menyelesaikan masalah. Seorang ahli adalah seseorang yang memiliki keahlian, kemahiran dan langkah agar dapat mengaplikasikannya untuk menyelesaikan persoalan atau memberikan nasihat. Dengan demikian, seorang ahli harus dapat melakukan tindakan berikut :

1. Identifikasi dan artikulasikan masalah.
2. Selesaikan masalah dengan cepat dan akurat.
3. Jelaskan solusinya.
4. Belajar dari pengalaman anda sendiri.
5. Restrukturisasi pengetahuan.
6. Untuk memecahkan aturan.
7. Tentukan relevansinya (T. Sutojo, 2011 : 163).

Sistem pakar juga merupakan jenis KBS (sistem berbasis pengetahuan) yang menggunakan pengetahuan untuk menyelesaikan tugas seorang pakar. Sedangkan KBS sendiri merupakan sistem berbasis pengetahuan yang lebih luas dan lebih luas dari pada sistem pakar. Sistem pakar membutuhkan pengetahuan, beberapa pemahaman tentang penggunaan, dan kemampuan untuk berkomunikasi dengan pengguna. Ketiga hal ini sering disebut sebagai *knowledge base, human-machine interface*. Karakteristik sistem pakar adalah sebagai berikut :

1. Terbatas untuk bidang tertentu.
2. Dapat memberikan justifikasi untuk data yang tidak lengkap atau tidak akurat.

3. Dapat menyatakan sejumlah alasan dengan cara yang dapat dimengerti.
4. Berdasarkan aturan atau regulasi tertentu.
5. Didesain untuk pengembangan bertahap.
6. Hasilnya tergantung pada dialog dengan pengguna.
7. Pisahkan basis pengetahuan dan mesin inferensi.

Kategori tugas sistem pakar yang dapat diselesaikan oleh sistem pakar termasuk :

1. Interpretasi - untuk menarik kesimpulan atau deskripsi dari satu set atau deskripsi dari satu set dari bahan mentah.
2. Prediksi - memprediksi kemungkinan konsekuensi dari situasi tertentu.
3. Diagnostik - menentukan penyebab kerusakan dalam situasi sulit berdasarkan gejala yang diamati.
4. Desain - mendefinisikan konfigurasi komponen sistem yang konsisten dengan tujuan kinerja.

2.3 Metode Runut maju (*Forward Chaining*).

Proses ini diulangi sampai hasilnya ditemukan. Rantai lurus juga bisa disebut metode inferensi yang dimulai dengan serangkaian anda cari. Yang pertama kali dimulai dengan memasukkan data dan kemudian mengarah ke kesimpulan. Contohnya adalah aturan “jika-maka”, yang dapat digunakan untuk menentukan suatu masalah dengan menggunakan metode ini. Penggunaan pohon keputusan ini dimaksudkan untuk memudahkan dalam mendeskripsikan dan memahami pengetahuan (Abdullah, 2018). Rantai lurus adalah sekelompok kesimpulan ganda, di mana kesimpulan adalah proses yang menghasilkan informasi atau keputusan berdasarkan fakta yang diketahui. Dalam pengambilan keputusan berdasarkan fakta yang diketahui menggunakan inferensi, terdapat beberapa aturan inferensi, yaitu:

- Modus ponens (*law of detachment*)

Aturan ini didasarkan pada tautologi $(p \wedge (p \rightarrow q)) \rightarrow q$, di mana p dan $p \rightarrow q$ adalah hipotesis dan q adalah kesimpulan. Aturan ini juga dapat ditulis sebagai:

$$\begin{array}{c}
 p \rightarrow q \\
 p \\
 \hline
 \text{Jadi konklusi : } q
 \end{array}$$

- Modus tollen

Aturan ini didasarkan pada tautologi $[\sim q \wedge (p \rightarrow q)] \rightarrow \sim p$ rule Aturan ini juga dapat ditulis sebagai:

$$\begin{array}{c}
 p \rightarrow q \\
 p \\
 \hline
 \text{Jadi konklusi : } \sim p
 \end{array}$$

- Silogisme hipotetis

Aturan ini didasarkan tautologi $[(p \rightarrow q) \wedge (q \rightarrow r)] \rightarrow (p \rightarrow r)$ rule ini juga dapat ditulis dengan cara :

$$\begin{array}{c}
 p \rightarrow q \\
 q \rightarrow r \\
 \hline
 \text{Jadi konklusi : } p \rightarrow r
 \end{array}$$

- Silogisme disjungtif

Aturan ini didasarkan tautologi $[(p \wedge q) \wedge \sim p] \rightarrow q$. rule ini juga dapat ditulis dengan cara :

$$\begin{array}{c}
 p \vee q \\
 \hline
 q \rightarrow r \\
 \hline
 \text{Jadi konklusi : } p \rightarrow r
 \end{array}$$

- Simplifikasi

Aturan ini didasarkan pada tautologi $(p \wedge q) \rightarrow p$. Rule ini juga dapat ditulis dengan cara :

$$\begin{array}{c}
 p \wedge q \\
 \hline
 \text{Jadi konklusi : } p
 \end{array}$$

- Penjumlahan

Aturan ini didasarkan pada tautologi $p \rightarrow (p \vee q)$. Rule ini juga dapat ditulis dengan cara :

$$\frac{p}{\text{Jadi konklusi : } p \vee r}$$

- Konjungsi

Aturan ini didasarkan tautologi $((p) \wedge (q)) \rightarrow (p \wedge q)$. Rule ini juga dapat ditulis dengan cara :

$$\frac{\begin{array}{c} p \\ q \end{array}}{\text{Jadi konklusi : } p \wedge q}$$

- Akurasi

Verifikasi tingkat akurasi yang dipertimbangkan terdiri dari penentuan persentase akurasi dalam proses pengklasifikasian data uji. Tingkat akurasi dihitung dengan rumus:

$$\text{Akurasi} = \frac{\sum \text{match}}{\sum \text{tp}} \times 100\%$$

$$\sum \text{match} = \text{jumlah klasifikasi yang benar}$$

$$\sum \text{tp} = \text{jumlah data testing}$$

Rantai depan adalah fakta untuk disimpulkan dari fakta ini. Dengan kata lain, proses dimulai dari fakta melalui antarmuka fakta (*fact of reasoning*) ke objek. Cara ini disebut dengan memakai tata cara IF - THEN, ketika premis (IF) mengarah pada kesimpulan (THEN), bisa ditulis seperti ini:

THEN (konklusi)

Beberapa anggapan tentang penerapan langkah. Satu, adanya memasukkan semua data diterima ke dalam sistem pakar. Dua, hanya memasukkan beberapa yang penting termasuk data yang diperoleh ke sistem. Metode satu paling bisa dipakai jika sistem dihubungkan ke proses dan terima semua dari database. Misalnya, seperti dua metode yang dijelaskan di atas, maka langkah-langkah berikut diambil berdasarkan metode ini:

R1 : IF A and C, THEN B

R2 : IF D and C, THEN F

R3 : IF B and E, THEN F

R4 : IF B, THEN C

R5 : IF F, THEN G

Kedua jenis strategi tersebut akan menghasilkan kesimpulan. Namun, efektifitas tergantung pada keadaan masalah yang sedang dipertimbangkan: jika masalah memiliki premis yang kurang dari kesimpulan, maka strategi yang akan diusulkan oleh rantai balik (Windah Supartini, 2016). Produk inferensi yang menghubungkan masalah ke solusi disebut rangkaian. Metode rantai langsung memiliki karakteristik perencanaan, pemantauan, pengendalian, disajikan untuk masa depan, panduan data, penalaran bottom-up, bekerja untuk menemukan solusi yang sesuai dengan fakta, dan pencarian luas-pertama semuanya menjadi lebih mudah. *Forward chaining* dapat disebut metode menemukan atau menyimpulkan kesimpulan berdasarkan data

2.4 Bahasa Pemrograman Java

Versi pertama bahasa pemrograman *java* dirilis pada akhir 1995, dan dalam beberapa bulan Java menjadi bahasa pemrograman *World Wide Web*. *Java* memiliki beberapa keunggulan dibandingkan bahasa pemrograman lainnya.

- a) *Java* bersifat lebih sederhana dan relatif mudah, *java* sebagian dimodelkan dalam C ++.
- b) *Java* adalah Bahasa Pemrograman Berorientasi Objek (OOP). Pemrograman berorientasi objek mengacu pada konsep pemrograman

yang memecahkan masalah dengan menyortir program menjadi objek yang berinteraksi satu sama lain.

- c) *Java* bersifat *multiplatform*, dapat diterjemahkan oleh juru bahasa *java* di berbagai sistem operasi.
- d) *Java* adalah *multithread*, *thread* adalah proses di mana program dapat berjalan secara bersamaan. Ini berarti *java* dapat berjalan di banyak proses hampir secara bersamaan.

2.5 Aplikasi Berbasis Mobile (Android)

Aplikasi android Ini adalah aplikasi seluler yang perlu diinstal di perangkat target. Aplikasi ini dapat disebut sebagai aplikasi platform karena aplikasi ini harus dikembangkan dan disusun secara khusus untuk setiap platform seluler. (Tawfiq Ramadhan, 2014)

Menurut Muladi (2010: 5), Android merupakan bagian dari perangkat lunak seluler yang mencakup sistem operasi, *middleware*, dan aplikasi yang dirilis oleh *Google*. Muladi (2010: 8) mendeskripsikan arsitektur android sebagai berikut:

- A. Aplikasi lapisan aplikasi ini nantinya berisi aplikasi yang dikembangkan oleh *programmer*.
 - B. Platform Aplikasi Pengembang, aplikasi memiliki akses penuh ke API platform yang sama yang digunakan oleh aplikasi inti.
 - C. Arsitektur aplikasi dirancang agar komponen aplikasi dapat digunakan kembali dengan mudah.
 - D. *Library Android* menyertakan *library C / C ++* yang digunakan oleh berbagai komponen sistem android. Pengembang aplikasi android dapat mengakses kapabilitas pustaka melalui kerangka aplikasi android.
 - E. Waktu proses android terdiri dari sekumpulan pustaka inti yang menyediakan banyak fungsi yang sama dengan pustaka inti bahasa pemrograman *java*.
 - F. Kernel Android *Linux* bukanlah *Linux*, tetapi Android dibangun di atas kernel *Linux 2.6*, sehingga Anda dapat mempercayai keandalannya.
- (Tawfiq Ramadhan, 2014)

2.6 Mata Rabun

Mata adalah instrumen sensorik manusia yang terus-menerus menyesuaikan dengan jumlah cahaya yang datang, memfokuskan perhatian pada objek yang dekat dan jauh dan menciptakan gambar yang terus menerus yang segera dikirim ke otak. Penyakit mata rabun adalah suatu kondisi di mana gambar padat tidak terbentuk di retina, tetapi terjadi ketidakseimbangan dalam sistem. Sistem penglihatan di mata untuk membuat gambar kabur. Berikut ciri-ciri penyakit mata rabun, yaitu:

1. Mata *Myopi*

Rabun jauh, juga dikenal sebagai *miopia*, adalah suatu kondisi penglihatan di mana anda tidak dapat melihat objek di kejauhan, tetapi anda dapat melihat objek di sekitar secara jelas dengan menggunakan alat bantu. Jika anda mengalami rabun jauh, anda akan kesulitan melihat objek yang jauh.

Tanda dan gejala rabun jauh adalah:

- Penglihatan kabur saat melihat objek dari kejauhan.
- Untuk melihat lebih baik, Anda perlu menyipitkan mata atau menutup sebagian kelopak mata.
- Sakit kepala karena mata lelah.
- Masalah penglihatan saat mengemudi, terutama di malam hari.
- Mata juling konstan.
- Anda harus duduk dekat dengan TV, layar lebar, atau di depan kelas.
- Tampak lupa pada objek yang jauh.
- Berkedip berlebihan.
- Pemakaian komputer lebih dari 3 jam sehari.

2. Mata *Hipermetropi*

Rabun jauh atau *hiperopia* adalah gangguan penglihatan dekat. Pada penderita *hiperopia*, objek yang jauh terlihat jelas, tetapi objek yang dekat tidak jelas atau buram. Bayi dan anak kecil cenderung mengalami *hiperopia*, tetapi penglihatan mereka tidak kabur. Tidak ada yang perlu dikhawatirkan, karena seiring berkembangnya organ mata, penglihatan akan menjadi normal.

Penderita hipermetropi akan mengalami gejala berikut ini:

- Penglihatan tidak fokus saat melihat objek dekat.
- Anda harus menyipitkan mata untuk melihat sesuatu yang lebih jelas.
- Seringkali berair jika Anda membaca dalam waktu lama.
- Kelelahan mata atau sakit kepala setelah penglihatan jarak dekat dalam waktu lama, seperti saat menulis, membaca, atau menggunakan komputer.

3. Mata Katarak Senilis

Katarak adalah segala kondisi opasitas lensa yang mungkin diakibatkan oleh hidrasi (kekurangan penambahan cairan) ke lensa, denaturasi protein lensa, atau keduanya. Lima puluh satu persen (51%) kebutaan disebabkan oleh katarak. Katarak pikun adalah jenis katarak yang paling umum dan ada penderitanya.

Penderita Katarak akan mengalami gejala berikut ini:

- Penglihatan itu mendung seperti kabut.
- Warna di sekitarnya tampak pudar.
- Pertimbangkan bagian cahaya di sekitar lampu.
- Tampilan ganda.
- Penglihatan lebih nyaman di malam hari.
- Ubah ukuran kacamata Anda sesering mungkin.

2.7 Riset Terkait

Ada beberapa penelitian terkait yang menggunakan sistem pakar Forward Chaining, yaitu:

- a) Pengembangan sistem pakar diagnosis asidosis tubulus ginjal menggunakan direct chain confidence factor oleh Rahmi Ras Fanny, dkk. Memudahkan identifikasi gejala asidosis tubulus ginjal. Nilai kepercayaan dalam diagnosis asidosis tubulus ginjal yang disebabkan oleh sistem ini sama dengan yang dihasilkan perhitungan manual menggunakan metode faktor kepercayaan. Sehingga hasilnya sesuai dengan kalkulasi yang diharapkan.
- b) Aplikasi sistem pakar untuk mendeteksi kerusakan laptop / PC dengan metode direct chaining menggunakan bahasa pemrograman Java yang dikembangkan oleh Hasanah, dkk yaitu dengan menggunakan aplikasi sistem pakar khususnya aplikasi pendeteksi kerusakan laptop / PC dapat

memudahkan orang dalam melakukan perbaikan. kerusakan pada perangkat komputer mereka.

- c) Sistem Pakar Berbasis Web Dengan Metode Forward Chaining Dalam Mendiagnosis Dini Penyakit Tuberkulosis di Jawa Timur oleh Winda Supartini, dkk merupakan sistem pakar untuk diagnosa dini TB dengan menggunakan Metode Rantai Langsung Berbasis Internet, yang sangat berguna untuk diagnosa dari TB. berdasarkan gejala yang dikeluhkan pasien.
- d) Sistem Pakar Berbasis Web Menggunakan Metode *Forward Chaining* Untuk Mendiagnosa Penyakit Pulpa Dan Periapikal oleh Fadhilah, dkk adalah Metode rantai langsung dapat digunakan untuk membuat sistem pakar untuk diagnosis penyakit pulpa dan periapikal. Aplikasi Sistem Pakar Diagnosis Penyakit Pulp dan Penyakit Periapikal ini dapat membantu masyarakat pengguna dalam mendiagnosis Penyakit Pulp dan Penyakit Periapikal berdasarkan gejala yang dialaminya.
- e) Aplikasi Sistem Pakar Diagnosa Penyakit Pada Anak Bawah Lima Tahun Menggunakan Metode *Forward Chaining* oleh Bagus Fery Yanto, dkk merupakan aplikasi sistem pakar untuk mendiagnosis penyakit pada balita dengan metode rantai langsung, yang dapat dilakukan dalam tiga langkah. Langkah pertama adalah mengumpulkan data dan informasi dari MTBS dan wawancara. Tahap kedua adalah pembentukan aturan berdasarkan 18 penyakit dengan metode rantai langsung. Tahap ketiga adalah implementasi aplikasi sistem pakar berbasis android dengan fungsi diagnosa penyakit, riwayat diagnosa dan pengumpulan penyakit. Hasil evaluasi menunjukkan bahwa sistem memiliki tingkat akurasi 82% dengan menggunakan data dari 50 pengujian.
- f) Sistem Pakar Berbasis Android Untuk Diagnosa Penyakit Kulit Kucing Dengan Metode *Forward Chaining* oleh Siti Nurajizah, dkk adalah Aplikasi sistem pakar yang dibuat berbasis Android, sehingga pengguna dapat dengan mudah mengaksesnya kapanpun dan dimanapun hanya dengan membuka aplikasinya. Dengan bantuan sistem pakar ini, masyarakat umum dapat mengetahui berbagai gejala dan jenis penyakit, serta solusi yang diusulkan.

Semoga lebih banyak gejala, jenis penyakit dan solusinya dapat ditambahkan untuk mengembangkan aplikasinya lebih lanjut.

- g) Penerapan Metode *Forward Chaining* untuk Mendeteksi Penyakit THT oleh Wiwi Verina merupakan metode rantai langsung yang digunakan untuk mencari hasil penyakit pada organ THT (telinga, hidung dan tenggorokan). Dengan demikian, dari hasil pencarian metode rantai langsung diperoleh 75 aturan untuk menentukan penyakit THT berdasarkan gejala yang dimasukkan oleh pengguna.

BAB III

METODE PENELITIAN

3.1 Tempat dan Waktu Penelitian

3.1.1 Tempat Penelitian

Lokasi diadakan di Praktek Spesialis Mata Dr. Zaldi Z. SpM yang berada di Jalan Bromo no 53 Medan, Sumatera Utara.

3.1.2 Waktu Dan Jadwal Pelaksanaan Penelitian

Pelaksanaan penelitian ini dilaksanakan pada semester genap tahun ajaran 2018/2019.

3.2 Bahan dan Alat Penelitian

Penelitian ini dilakukan dengan menggunakan alat pendukung seadanya untuk penelitian.

3.2.1 Perangkat Keras

Perangkat keras digunakan pada pengembangan sistem ini diperlukan sebagai berikut Laptop, Processor Intel(R) Celeron(R) CPU N3050 @1.60GHz, Installed Memory (RAM) 2.00 GB, dan Harddisk 500 GB.

3.2.2 Perangkat Lunak

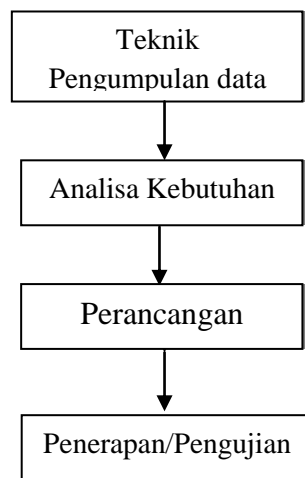
Software yang digunakan pada pengembangan sistem ini diperlukan sebagai berikut :

1. Operating System Windows 8.1 Pro 64 bit.
2. Bahasa Pemrograman *JAVA*
3. *Android Studio*
4. *Android Emulator*

3.3 Cara Kerja

3.3.1 Perencanaan

Perencanaan dalam penelitian ini dilakukan melalui tahapan sebagai berikut:



Gambar 3.1 Tahap – Tahap Perencanaan

3.3.2 Teknik Pengumpulan Data

Penelitian ini menggunakan metode pengumpulan data sebagai berikut:

1. Penelitian Kepustakaan (*Library Research*)
2. Studi Literatural
3. Wawancara

Berikut data yang didapatkan dari hasil Penelitian Kepustakaan (*Library Research*), Studi Literatural dan Wawancara yaitu :

1. Data gejala

Data gejala yang digunakan dalam sistem pakar penyakit mata rabun memiliki 19 gejala. Data untuk gejala ini bisa jadi ditunjukkan pada tabel 3.1 di bawah ini:

Kode Gejala	Nama Gejala
G01	Penglihatan kabur saat melihat objek dari kejauhan
G02	Untuk melihat lebih baik, Anda perlu menyipitkan mata atau menutup sebagian kelopak mata
G03	Sakit kepala karena mata lelah
G04	Masalah penglihatan saat mengemudi, terutama di malam hari
G05	Mata dicipitkan

G06	Harus duduk dekat dengan TV, layar lebar, atau di depan kelas
G07	Tampak tidak sadar pada objek yang jauh
G08	Berkedip berlebihan
G09	Pemakaian komputer lebih dari 3 jam sehari
G10	Penglihatan tidak fokus ketika melihat objek dekat
G11	Seringkali berair jika membaca dalam waktu lama
G12	Kelelahan mata atau sakit kepala setelah penglihatan jarak dekat dalam waktu lama, seperti saat menulis, membaca, atau menggunakan computer
G13	Penglihatan itu mendung seperti kabut
G14	Warna di sekitarnya tampak pudar
G15	Pertimbangkan bagian cahaya di sekitar lampu
G16	Tampilan ganda
G17	Penglihatan lebih nyaman di malam hari
G18	Ubah ukuran kacamata Anda sesering mungkin

Tabel 3.1 Data Gejala

2. Data penyakit

Banyaknya penyakit yang diproses pada aplikasi penyakit mata rabun ini terdapat 3 jenis penyakit. Data penyakit ini dapat dilihat pada tabel 3.2:

Kode Penyakit	Nama Penyakit
P01	Rabun Jauh (<i>Myopi</i>)
P02	Rabun Dekat (<i>Hipermetropi</i>)
P03	Katarak <i>Senilis</i>

Tabel 3.2 Data Penyakit

Basis pengetahuan tersebut dapat dilihat pada tabel 3.3:

Kode Gejala (G)	Kode Penyakit (P)		
	01	02	03
G01	*		*
G02	*	*	
G03	*	*	
G04	*	*	*
G05	*	*	
G06	*		
G07	*		
G08	*	*	
G09	*	*	
G10		*	
G11		*	
G12		*	
G13	*		*
G14			*
G15			*
G16			*
G17	*		*
G18	*		*

Tabel 3.3 Keputusan Penyakit Mata Rabun

3.3.3 Analisa Kebutuhan

Dengan penyakit mata *miopia*, kebanyakan orang terkena rabun jauh diprediksi *miopia*. *Miopia* biasanya dimulai pada usia muda dan seseorang berisiko lebih tinggi jika orang tuanya juga memiliki riwayat *miopia*. Dalam banyak kasus, *miopia* dapat sembuh dengan sendirinya di awal masa dewasa,

tetapi terkadang miopia berkembang menjadi penuaan seiring bertambahnya usia. Mirip dengan rabun dekat atau *hyperopia*.

Pada dasarnya penderita umum penyakit mata rabun ini tidak menyadari akan penyakit yang dia alami. Namun untuk menentukan penyakit mata rabun dekat atau rabun jauh bukanlah mudah. Banyaknya penderita mata rabun dengan merasakan gejala - gejala yang penderita rasakan setiap harinya dan mata rabun ini juga dapat mengganggu pekerjaan ataupun kegiatan sehari-hari oleh penderita itu sendiri. Dan tidak butuh waktu lama untuk mengobati penyakit mata rabun tersebut dikarenakan sudah mengetahui apa penyakit dan bagaimana cara mengobati secara sederhana seperti apa yang diharapkan oleh si penderita mata rabun.

3.3.4 Perancangan

Tahap ini adalah inti teknis dari proses pengembangan perangkat lunak.

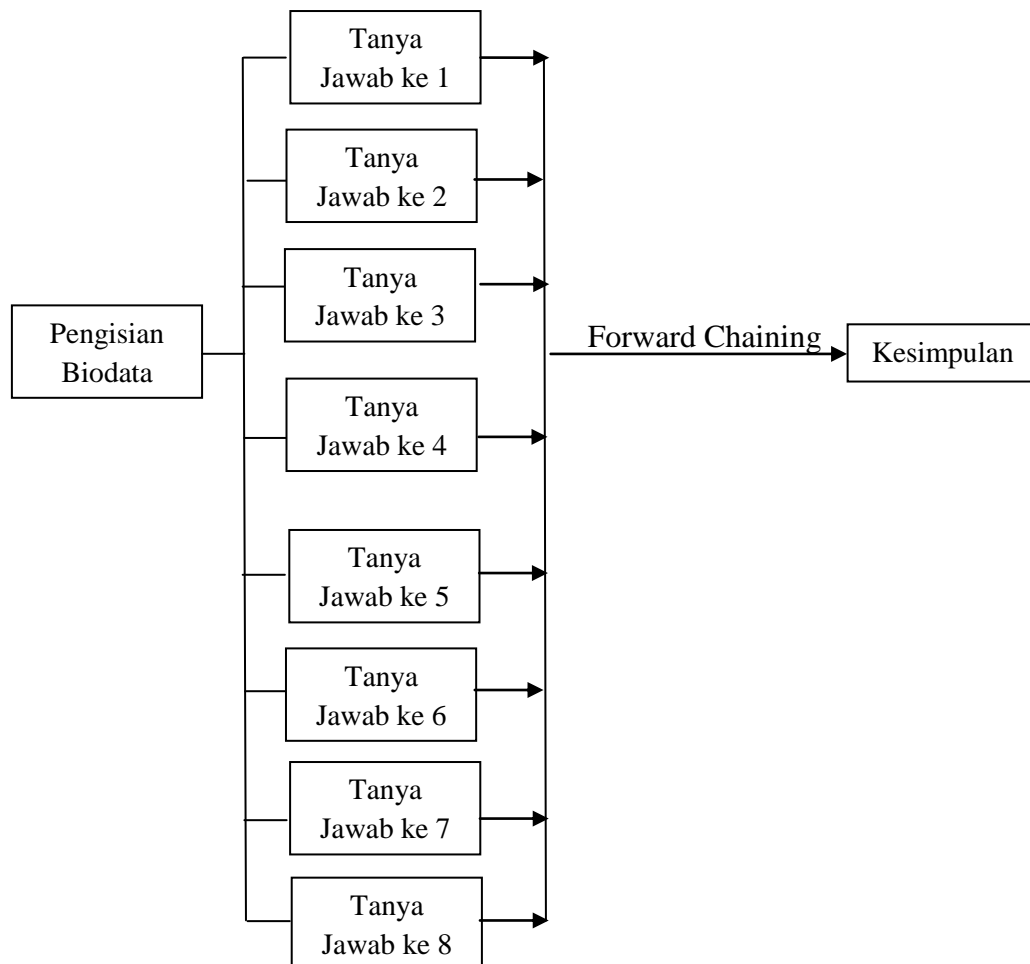
USER	SISTEM
<ul style="list-style-type: none"> • Buka aplikasi • Melakukan registrasi biodata user. • Mengisi rules dan pertanyaan dari gejala yang dirasakan. 	<ul style="list-style-type: none"> • Menampilkan menu registrasi biodata user. • Memproses data • Menampilkan form rules dan pertanyaan – pertanyaan gejala yang dirasakan. • Menampilkan hasil dari proses penelusuran

Tabel 3.4 Keterangan Perancangan Diagram

- **Forward Chaining**

Penggunaan metode *forward chaining* pada aplikasi tersebut ada pada proses pengumpulan jawaban dari pertanyaan yang digunakan sebagai fakta dan nantinya akan digunakan dalam proses penentuan kepribadian. Proses

pengumpulan fakta dan penggunaan *forward chaining* dalam aplikasi dapat dilihat dari gambar berikut ini



Gambar 3.2 *Forward Chaining System*

Pada gambar diatas dijelaskan bahwa metode *forward chainning* pada aplikasi ini dimulai dari pengumpulan fakta berupa jawaban dari pertanyaan pertama sampai terakhir yang diberikan oleh aplikasi kepada pengguna yang nanti akan digunakan untuk menarik kesimpulan tentang jenis penyakit dari pengguna.

3.3.5 Pengujian

Pengujian sistem ditujukan untuk menguji kinerja sistem serta mengetahui secara cepat dan lebih efektif jenis penyakit mata rabun. Pengujian dilakukan oleh

pengguna awam maupun penguji kualitas dengan cara menggunakan sistem tersebut secara langsung. Sistem akan diujikan kepada beberapa orang peserta yang dipilih secara acak pada setiap pengujian. Pengujian akan dimulai dari penjelasan tentang program kepada orang tersebut dan mereka akan diminta menginput data dan mengisi pertanyaan yang diberikan oleh sistem dengan gejala mereka rasakan. Langkah pertama user membuka aplikasi, kemudian melakukan registrasi biodata diri untuk masuk ke dalam sistem lalu klik simpan. Secara otomatis data tersebut tersimpan kedalam data base. Selanjutnya user menjawab beberapa pertanyaan di dalam aplikasi dengan menjawab True and False. Setelah user menjawab pertanyaan tersebut, user mengklik tombol lanjutkan. Laporan hasil jawaban yang telah diisi oleh user akan terlihat secara otomatis. kemudian klik tombol keluar untuk keluar dari program.

3.3.6 Penerapan / Penggunaan

Penerapan memudahkan anda untuk menentukan jenis penyakit mata rabun berdasarkan gejala yang dia rasakan. Menggunakan sistem pakar dan menerapkan metode rantai langsung yang dirancang untuk memudahkan pasien dalam mengidentifikasi jenis penyakit mata rabun. Dan sebagai pengetahuan bagi para pengguna internet tentang pentingnya menjaga kesehatan mata dari segi gejala mata miopia pada sistem.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pembahasan

4.1.1 Analisis Data

Penggunaan metode *forward chaining* pada aplikasi tersebut ada pada proses pengumpulan jawaban dari pertanyaan yang digunakan sebagai fakta dan nantinya akan digunakan dalam proses penentuan kepribadian. Proses pengumpulan fakta dan penggunaan *forward chaining* dalam contoh kasus dapat dilihat dalam aplikasi tersebut.

4.1.2 Representasi Data

Data-data tersebut di dapatkan melalui beberapa tahap pengumpulan data yaitu wawancara bersama Dr. Zaldi Z, SpM di lokasi praktek beliau Jl. Bromo 63 A/B (Samping Apotek Bromo), buku dan jurnal yang berhubungan dengan penyakit mata rabun.

1. Data gejala

Data gejala yang digunakan dalam sistem pakar penyakit mata rabun memiliki 19 gejala. Data untuk gejala ini bisa jadi ditunjukkan pada tabel 4.1 di bawah ini :

Kode Gejala	Nama Gejala
G01	Penglihatan kabur saat melihat objek dari kejauhan
G02	Untuk melihat lebih baik, Anda perlu menyipitkan mata atau menutup sebagian kelopak mata

G03	Sakit kepala karena mata lelah
G04	Masalah penglihatan saat mengemudi, terutama di malam hari
G05	Mata dicipitkan
G06	Harus duduk dekat dengan TV, layar lebar, atau di depan kelas
G07	Tampak tidak sadar pada objek yang jauh
G08	Berkedip berlebihan
G09	Pemakaian komputer lebih dari 3 jam sehari
G10	Penglihatan tidak fokus ketika melihat objek dekat
G11	Seringkali berair jika membaca dalam waktu lama
G12	Kelelahan mata atau sakit kepala setelah penglihatan jarak dekat dalam waktu lama, seperti saat menulis, membaca, atau menggunakan computer
G13	Penglihatan itu mendung seperti kabut
G14	Warna di sekitarnya tampak pudar
G15	Pertimbangkan bagian cahaya di sekitar lampu
G16	Tampilan ganda
G17	Penglihatan lebih nyaman di malam hari
G18	Ubah ukuran kacamata Anda sesering mungkin

Tabel 4.1 Data Gejala

2. Data penyakit

Banyaknya penyakit yang diproses pada aplikasi penyakit mata rabun ini terdapat 3 jenis penyakit. Data penyakit ini dapat dilihat pada tabel 4.2:

Kode Penyakit	Nama Penyakit
P01	Rabun Jauh (<i>Myopi</i>)
P02	Rabun Dekat (<i>Hipermetropi</i>)
P03	Katarak Senilis

Tabel 4.2 Data Penyakit

Basis pengetahuan tersebut dapat dilihat pada tabel 4.3 :

Kode Gejala (G)	Kode Penyakit (P)		
	01	02	03
G01	*		*
G02	*	*	
G03	*	*	
G04	*	*	*
G05	*	*	
G06	*		
G07	*		
G08	*	*	
G09	*	*	
G10		*	
G11		*	
G12		*	
G13	*		*
G14			*
G15			*
G16			*
G17	*		*
G18	*		*

Tabel 4.3 Keputusan Penyakit Mata Rabun

4.1.3 Hasil Analisis Data

Verifikasi tingkat akurasi yang dipertimbangkan terdiri dari penentuan persentase akurasi dalam proses pengklasifikasian data uji. Tingkat akurasi dihitung dengan rumus:

$$\text{Akurasi} = \frac{\sum \text{match}}{\sum tp} \times 100\%$$

$\sum \text{match}$ = jumlah klasifikasi yang benar

$\sum tp$ = jumlah data testing

Sistem Pakar : Penasihat Penyakit Mata Rabun

Kasus : Tarmizi ingin berkonsultasi apakah dia mengalami penyakit mata rabun?

Basis pengetahuan dasar pengetahuan gejala dan skala dapat dilihat pada tabel berikut:

No	Kode	Gejala	Nilai Probabilitas
1	G01	Penglihatan kabur saat melihat objek dari kejauhan	1
2	G02	Untuk melihat lebih baik, Anda perlu menyipitkan mata atau menutup sebagian kelopak mata	1
3	G04	Masalah penglihatan saat mengemudi, terutama di malam hari	1
4	G06	Harus duduk dekat dengan TV, layar lebar, atau di depan kelas	1
5	G07	Tampak tidak sadar pada objek yang jauh	1
6	G08	Berkedip berlebihan	1
7	G09	Pemakaian komputer lebih dari 3 jam sehari	1
8	G12	Kelelahan mata atau sakit kepala setelah penglihatan jarak dekat dalam waktu lama, seperti saat menulis, membaca, atau menggunakan computer	1
9	G15	Pertimbangkan bagian cahaya di sekitar lampu	1
10	G17	Penglihatan lebih nyaman di malam hari	1

Tabel 4.4 Basis Data Pengetahuan

Berdasarkan representasi pengetahuan untuk mendiagnosisnya, disusun daftar aturan seperti pada tabel berikut:

No	Aturan (<i>rule</i>)
1	Jika G01 = Benar, maka Nilai Probabilitas = 1 Jika G01 = Tidak, maka Nilai Probabilitas = 0
2	Jika G02 = Benar, maka Nilai Probabilitas = 1 Jika G02 = Tidak, maka Nilai Probabilitas = 0
3	Jika G04 = Benar, maka Nilai Probabilitas = 1 Jika G04 = Tidak, maka Nilai Probabilitas = 0
4	Jika G06 = Benar, maka Nilai Probabilitas = 1 Jika G06 = Tidak, maka Nilai Probabilitas = 0
5	Jika G07 = Benar, maka Nilai Probabilitas = 1 Jika G07 = Tidak, maka Nilai Probabilitas = 0
6	Jika G08 = Benar, maka Nilai Probabilitas = 1 Jika G08 = Tidak, maka Nilai Probabilitas = 0
7	Jika G09 = Benar, maka Nilai Probabilitas = 1 Jika G09 = Tidak, maka Nilai Probabilitas = 0
8	Jika G12 = Benar, maka Nilai Probabilitas = 1 Jika G12 = Tidak, maka Nilai Probabilitas = 0
9	Jika G15 = Benar, maka Nilai Probabilitas = 1 Jika G15 = Tidak, maka Nilai Probabilitas = 0
10	Jika G17 = Benar, maka Nilai Probabilitas = 1 Jika G17 = Tidak, maka Nilai Probabilitas = 0

Tabel 4.5 Aturan (*rule*)

Proses implementasi mesin inferensi yang menggunakan probabilitas akurasi *Forward Chaining*. Tarmizi mendiagnosis dengan menjawab pertanyaan tentang gejala berikut :

Variabel-variabel yang digunakan :

- a) Penglihatan kabur saat melihat objek dari kejauhan
- b) Untuk melihat lebih baik, Anda perlu menyipitkan mata atau menutup sebagian kelopak mata
- c) Masalah penglihatan saat mengemudi, terutama di malam hari
- d) Harus duduk dekat dengan TV, layar lebar, atau di depan kelas
- e) Tampak tidak sadar pada objek yang jauh
- f) Berkedip berlebihan
- g) Kelelahan mata atau sakit kepala setelah penglihatan jarak dekat dalam waktu lama, seperti saat menulis, membaca, atau menggunakan computer
- h) Pemakaian komputer lebih dari 3 jam sehari
- i) Pertimbangkan bagian cahaya di sekitar lampu
- j) Penurunan penglihatan dimalam hari

Hitungan Manual

P01 : Mata Rabun Jauh

$$G1 = 1$$

$$G2 = 1$$

$$G4 = 1$$

$$G6 = 1$$

$$G7 = 1$$

$$G8 = 1$$

$$G9 = 1$$

$$G12 = 0$$

$$G15 = 0$$

$$G17 = 1$$

$$\text{Akurasi} = \frac{1+1+1+1+1+1+1+0+0+1}{12} \times 100\% = 66,667$$

P02 : Mata Rabun Dekat

$$G1 = 0$$

$$G2 = 1$$

$$G4 = 1$$

$$G6 = 0$$

$$G7 = 0$$

$$G8 = 1$$

$$G9 = 1$$

$$G12 = 1$$

$$G15 = 0$$

$$G17 = 0$$

$$\text{Akurasi} = \frac{0+1+1+0+0+0+1+1+1+0+0}{9} \times 100\% = 55,55 \%$$

P03 : Mata Katarak Senilis

$$G1 = 1$$

$$G2 = 0$$

$$G4 = 1$$

$$G6 = 0$$

$$G7 = 0$$

$$G8 = 0$$

$$G9 = 0$$

$$G12 = 0$$

$$G15 = 1$$

$$G17 = 1$$

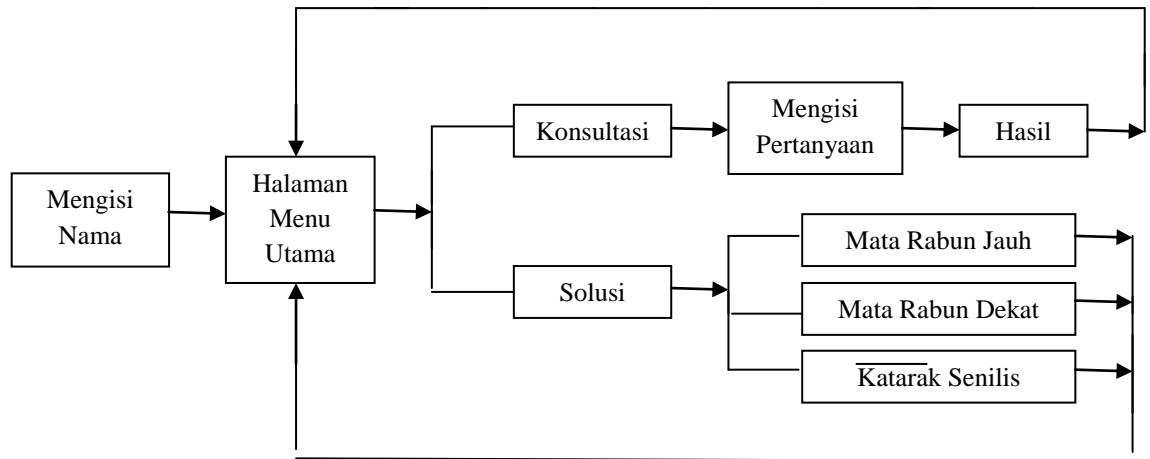
$$\text{Akurasi} = \frac{1+0+1+0+0+0+0+0+0+1+1}{8} \times 100\% = 50 \%$$

Tarmizi di diagnosis penyakit Mata Rabun Jauh dengan nilai akurasi 66,67 % atau 67 %.

4.1.4 Perancangan Sistem

1. Diagram

a. Perancangan Diagram Sistem



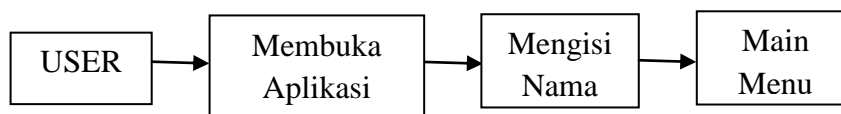
Gambar 4.1 Perancangan Diagram Sistem

b. Diagram Form Registrasi



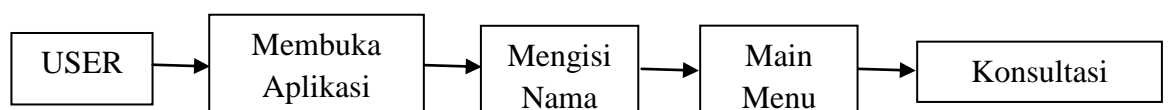
Gambar 4.2 Diagram Form Pendaftaran

c. Diagram Main Menu



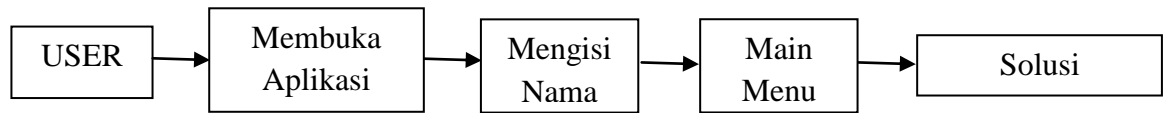
Gambar 4.3 Diagram Main Menu

d. Diagram Form Konsultasi



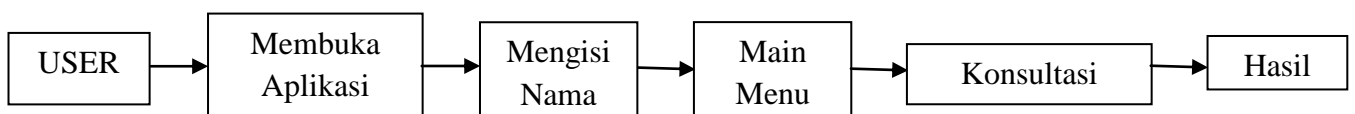
Gambar 4.4 Diagram Form Konsultasi

e. *Diagram Form Solusi*



Gambar 4.5 *Diagram Form Solusi*

f. *Diagram Form Hasil*

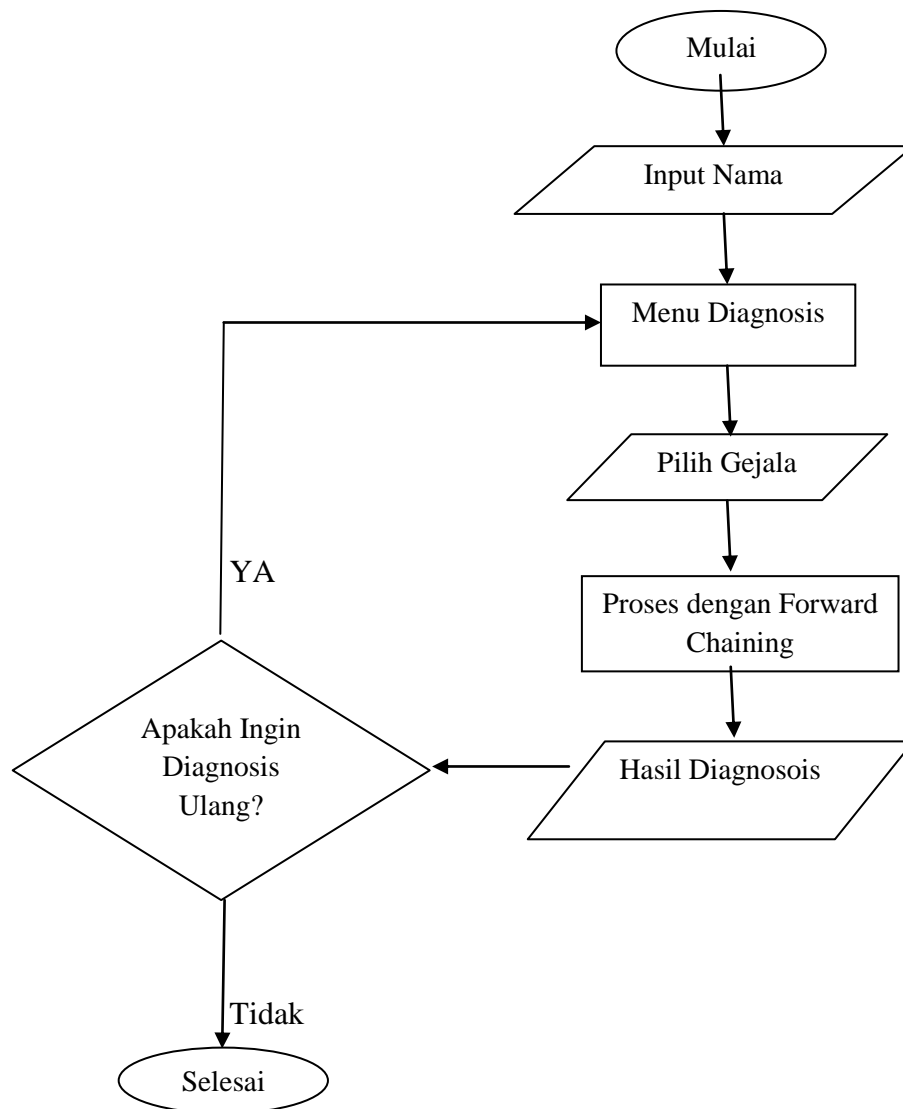


Gambar 4.6 *Diagram Form Hasil Kosultasi*

2. Flowchart

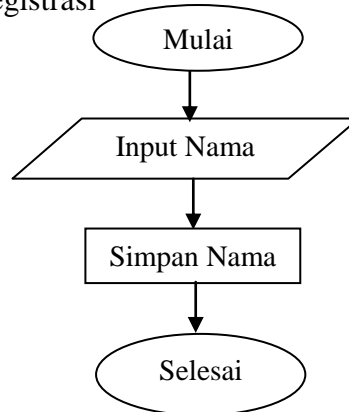
Diagram alir yang akan digunakan untuk mendiagnosis penyakit mata rabun sebagai berikut :

a) Perancangan *Flowchart* Sistem



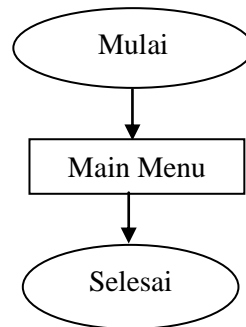
Gambar 4.7 Perancangan *Flowchart* Sistem

a. *Flowchart Form Registrasi*



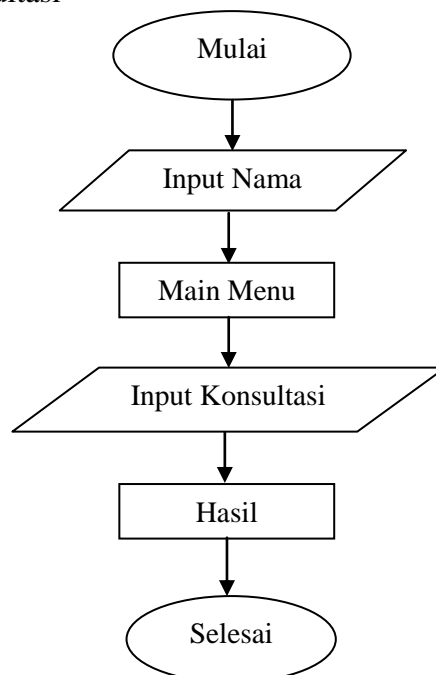
Gambar 4.8 Perancangan *Flowchart* Form Registrasi

b. *Flowchart Main Menu*

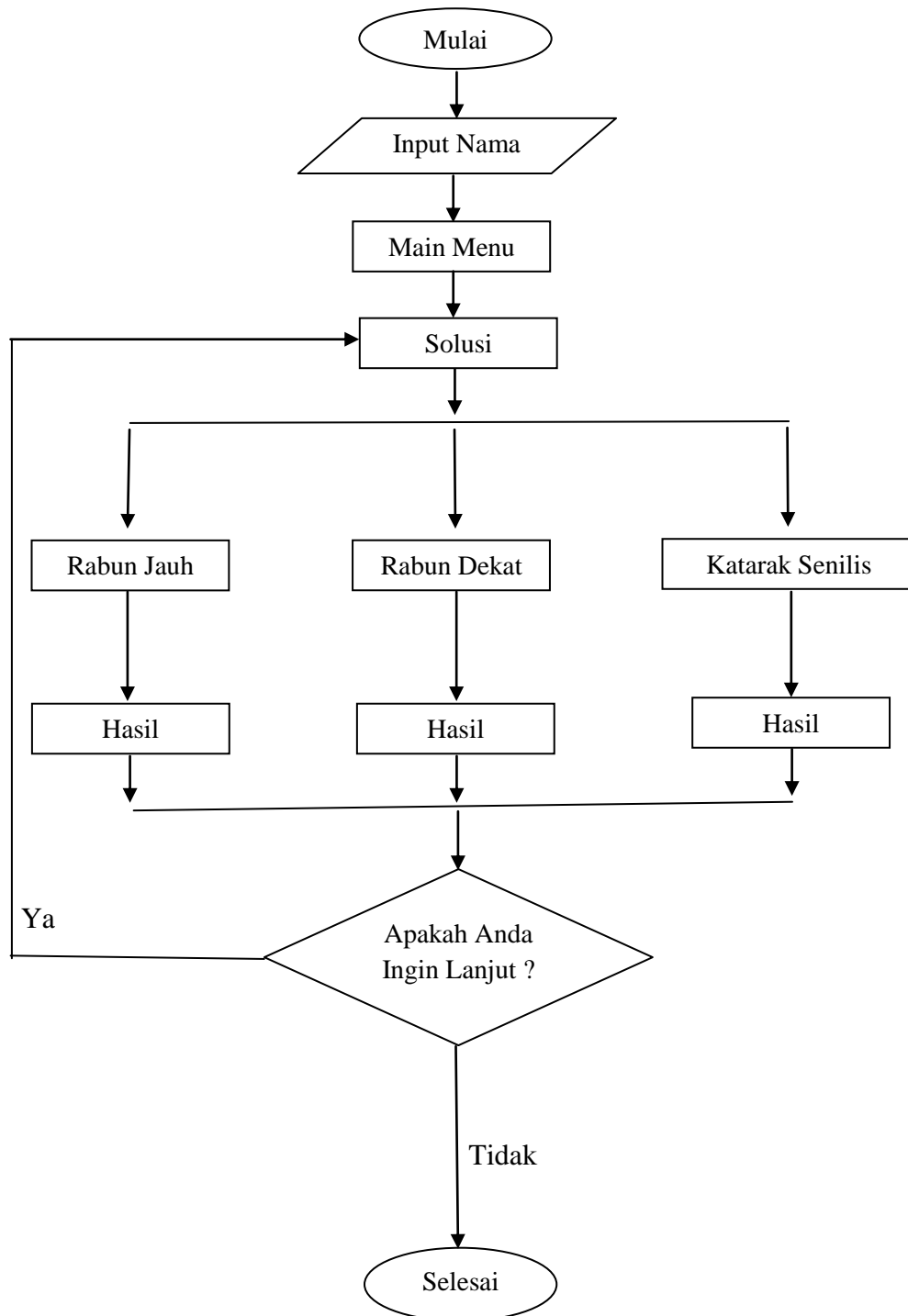


Gambar 4.9 Perancangan *Flowchart* Main Menu

c. *Flowchart Konsultasi*



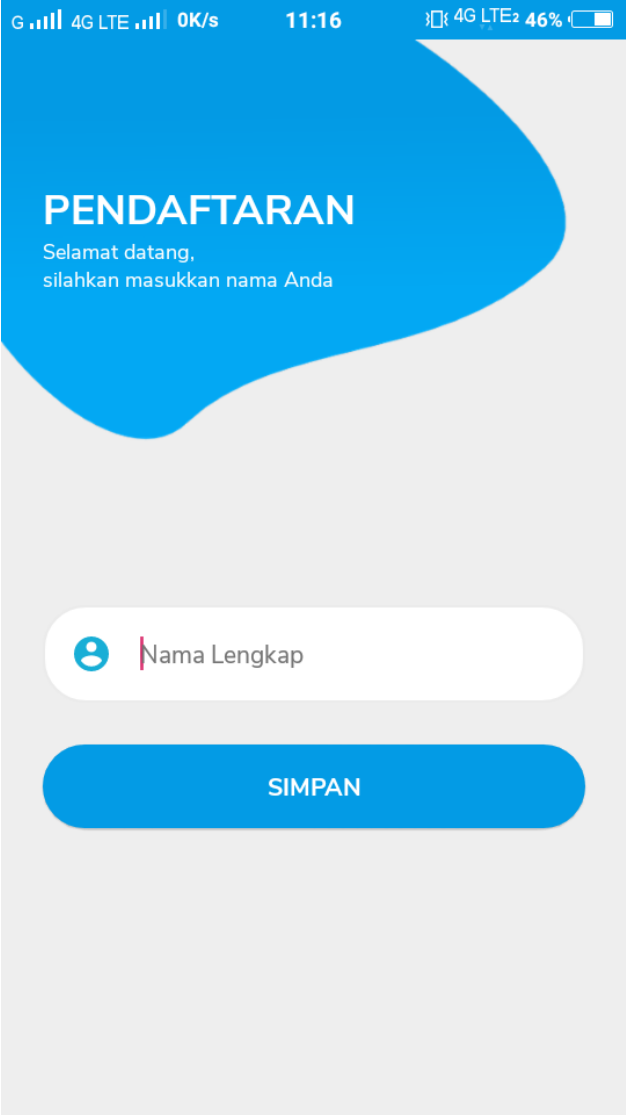
Gambar 4.10 Perancangan *Flowchart* Konsultasi

d. *Flowchart Solusi***Gambar 4.11** Perancangan *Flowchart Solusi*

3. Gambar Tampilan

a. Gambar Tampilan Form Registrasi

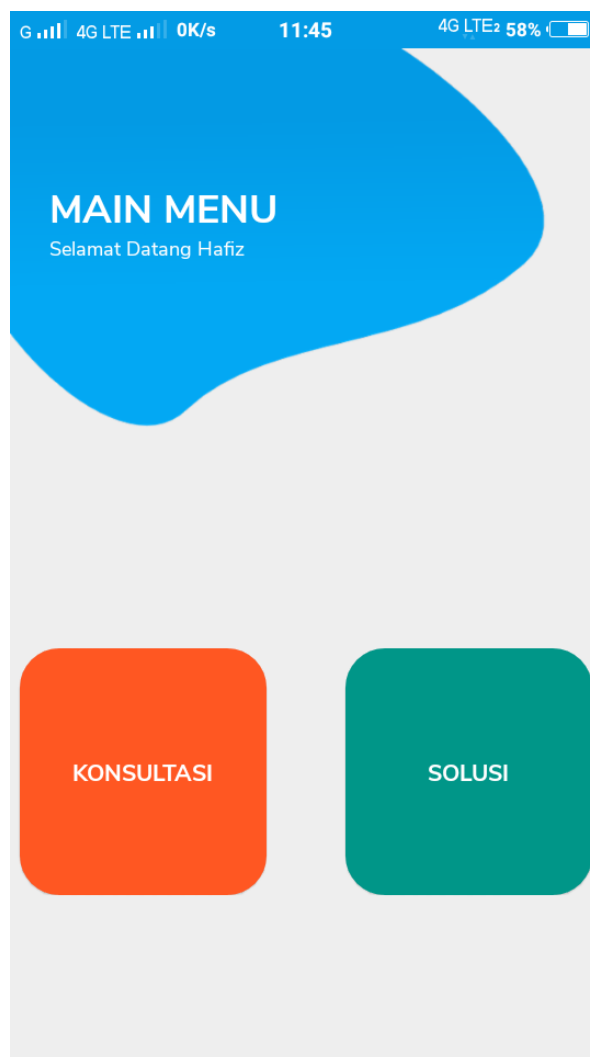
Form Registrasi merupakan form yang ada isian berupa pengisian nama lengkap serta tombol *simpan*. Jika *user* sudah mengisi nama lengkap sistem akan menampilkan form menu utama untuk bisa mengakses form konsultasi dan solusi, jika tidak maka form konsultasi dan solusi tidak akan bisa diakses, berikut gambar tampilan form registrasi seperti terlihat sebagai berikut:

The image is a screenshot of a mobile application's registration screen. At the top, there is a blue header with the word 'PENDAFTARAN' in white capital letters. Below the header, a light gray background contains a white rounded rectangular input field with a person icon on the left and the placeholder text 'Nama Lengkap'. Below the input field is a blue rounded rectangular button with the word 'SIMPAN' in white capital letters. The status bar at the top of the screen shows '4G LTE', '11:16', and '46%' battery.

Gambar 4.12 Gambar Tampilan *Form* Registrasi

b. Form Main Menu

Program yang baik adalah program yang memiliki integrasi dan berjalan menggunakan menu utama sebagai akses utama ke program, sehingga dapat memudahkan pemilik dan pengguna. Bentuk menu utama merupakan tampilan utama dari konten program, sedangkan tampilan menu utama pada aplikasi ini dapat melakukan operasi program secara optimal, dan juga dapat menggunakan kemampuan program yang ada. Menu utama aplikasi ditunjukkan di bawah ini:



Gambar 4.13 Gambar Tampilan *Form* Main Menu

c. Form Data Konsultasi

Form ini digunakan untuk menginput data konsultasi sesuai dengan yang sudah di tentukan oleh *user*.

DAFTAR DIAGNOSIS
Silahkan Pilih Diagnosis yang anda alami

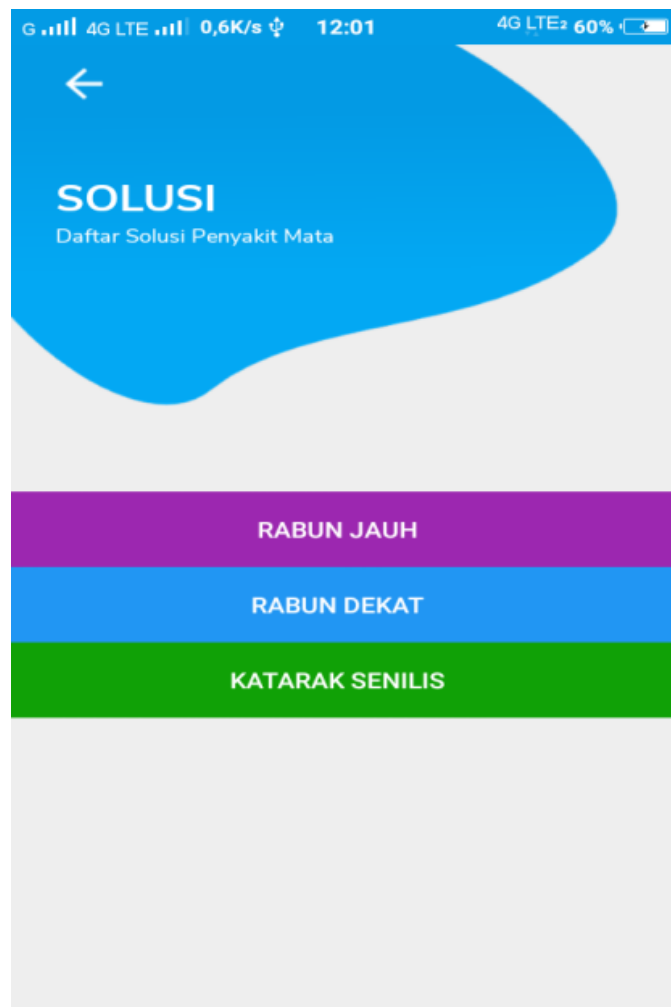
- 1 ☐ Penglihatan buram ketika melihat benda dalam jarak jauh
- 2 ☐ Harus menyipitkan mata atau sebagian menutup kelopak mata untuk melihat dengan jelas
- 3 ☐ Sakit kepala karena kelelahan mata
- 4 ☐ Kesulitan melihat saat berkendara, terutama pada malam hari
- 5 ☐ Terus-menerus menyipitkan mata
- 6 ☐ Harus duduk lebih dekat dengan televisi, layar lebar, atau di depan kelas
- 7 ☐ Tampak tidak menyadari adanya benda jauh
- 8 ☐ Mengedipkan mata secara berlebihan
- 9 ☐ Riwayat Pemakaian Komputer Lebih Dari 3 Jam Sehari
- 10 ☐ Penglihatan tidak fokus ketika melihat objek yang dekat
- 11 ☐ Sering Berair Jika Membaca Lama
- 12 ☐ Mata lelah atau sakit kepala usai melihat pada jarak dekat dalam waktu lama, misalnya menulis, membaca atau menggunakan Komputer
- 13 ☐ Pandangan kabur seperti berkabut
- 14 ☐ Warna di sekitar terlihat memudar
- 15 ☐ Melihat lingkaran cahaya di sekeliling sumber cahaya
- 16 ☐ Pandangan ganda
- 17 ☐ Penglihatan Lebih Nyaman Dimalam Hari
- 18 ☐ Sering mengganti ukuran kacamata

DIAGNOSIS PENYAKIT

Gambar 4.14 Gambar Tampilan *Form* Konsultasi

d. Form Solusi

Form ini digunakan untuk memberikan pengetahuan (*knowledge*) kepada *user* tentang bagaimana cara mengobati dan cara mencegah di dalam *data* solusi sesuai dengan yang sudah di tentukan oleh *user*.



Gambar 4.15 Gambar Tampilan *Form* Solusi

4.2 Hasil

4.2.1 Pengujian

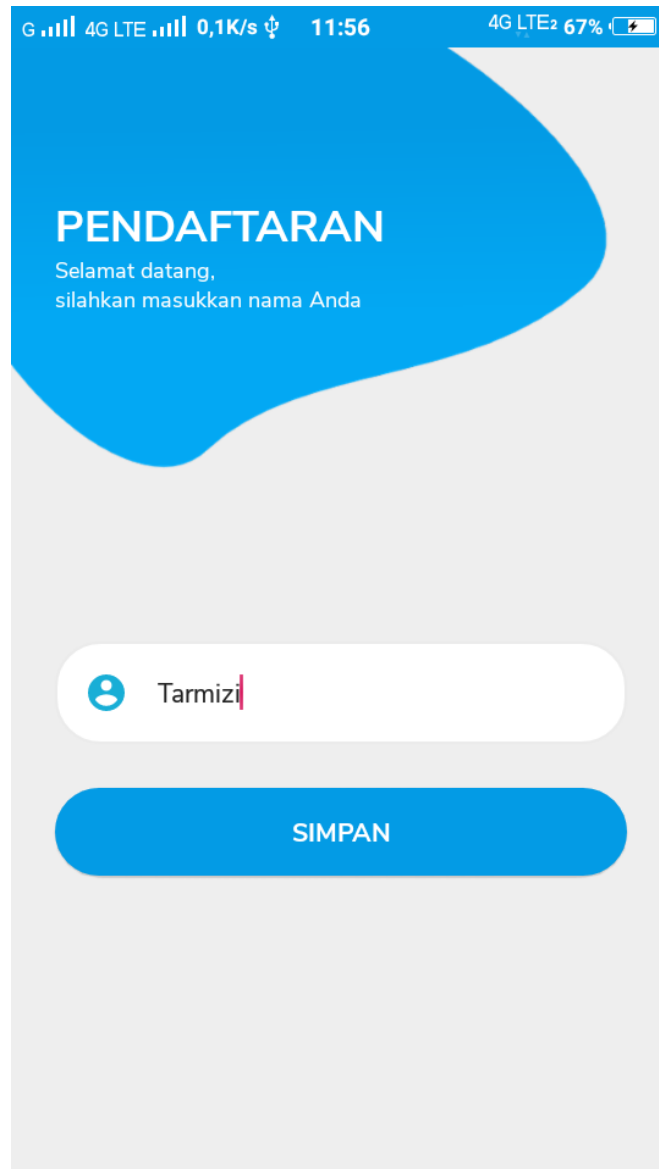
Setelah perancangan dan pembuatan sistem, dilakukan pengujian. Tujuan pengujian adalah untuk melihat bagaimana sistem yang dibangun memenuhi harapan. Proses pengumpulan fakta dan penggunaan aplikasi menggunakan direct chaining pada contoh dapat dilihat di bawah ini:

1. Mata Rabun Jauh

a. Menginput Data Registrasi

Pertama kita akan menginputkan data registrasi yang diberikan oleh sistem sebagai berikut :

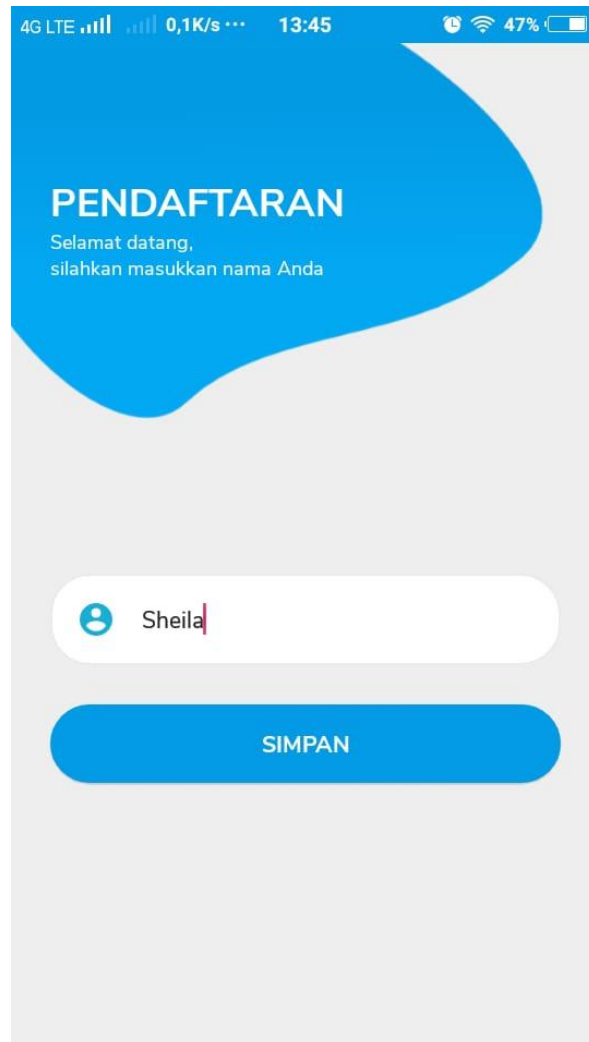
Kasus ke -1



The screenshot displays a mobile application interface for registration. At the top, a blue header contains the title "PENDAFTARAN" in white capital letters. Below the title, a smaller white text reads "Selamat datang, silahkan masukkan nama Anda". A white text input field with rounded corners is positioned below the text, containing the name "Tarmizi" and a red cursor. A blue button with the text "SIMPAN" in white capital letters is located below the input field. The status bar at the top of the phone shows "4G LTE", "0,1K/s", "11:56", and "67%" battery.

Gambar 4.16 Input Data Registrasi Rabun Jauh


Kasus ke -2



4G LTE 0,1K/s ... 13:45 47%

PENDAFTARAN

Selamat datang,
silahkan masukkan nama Anda

 Sheila

SIMPAN

Gambar 4.17 Input Data Registrasi Kasus ke 2 Rabun Jauh

b. Menginput Data Diagnosis

Kemudian kita akan menginputkan data konsultasi sesuai yang sudah ditentukan oleh sistem.

Kasus ke -1

DAFTAR DIAGNOSIS
Silahkan Pilih Diagnosis yang anda alami

- 1 ☒ Penglihatan buram ketika melihat benda dalam jarak jauh
- 2 ☒ Harus menyipitkan mata atau sebagian menutup kelopak mata untuk melihat dengan jelas
- 3 ☐ Sakit kepala karena kelelahan mata
- 4 ☒ Kesulitan melihat saat berkendara, terutama pada malam hari
- 5 ☐ Terus-menerus menyipitkan mata
- 6 ☒ Harus duduk lebih dekat dengan televisi, layar lebar, atau di depan kelas
- 7 ☒ Tampak tidak menyadari adanya benda jauh
- 8 ☒ Mengedipkan mata secara berlebihan
- 9 ☒ Riwayat Pemakaian Komputer Lebih Dari 3 Jam Sehari
- 10 ☐ Penglihatan tidak fokus ketika melihat objek yang dekat
- 11 ☐ Sering Berair Jika Membaca Lama
- 12 ☒ Mata lelah atau sakit kepala usai melihat pada jarak dekat dalam waktu lama, misalnya menulis, membaca atau menggunakan Komputer
- 13 ☐ Pandangan kabur seperti berkabut
- 14 ☐ Warna di sekitar terlihat memudar
- 15 ☒ Melihat lingkaran cahaya di sekeliling sumber cahaya
- 16 ☐ Pandangan ganda
- 17 ☒ Penglihatan Lebih Nyaman Dimalam Hari
- 18 ☐ Sering mengganti ukuran kacamata

DIAGNOSIS PENYAKIT

Gambar 4.18 Input Data Konsultasi Rabun Jauh

Kasus ke -2

DAFTAR DIAGNOSIS
Silahkan Pilih Diagnosis yang anda alami

- 1 ☒ Penglihatan buram ketika melihat benda dalam jarak jauh
- 2 ☒ Harus menyipitkan mata atau sebagian menutup kelopak mata untuk melihat dengan jelas
- 3 ☒ Sakit kepala karena kelelahan mata
- 4 ☒ Kesulitan melihat saat berkendara, terutama pada malam hari
- 5 ☒ Terus-menerus menyipitkan mata
- 6 ☒ Harus duduk lebih dekat dengan televisi, layar lebar, atau di depan kelas
- 7 ☒ Tampak tidak menyadari adanya benda jauh
- 8 ☒ Mengedipkan mata secara berlebihan
- 9 ☒ Riwayat Pemakaian Komputer Lebih Dari 3 Jam Sehari
- 10 ☐ Penglihatan tidak fokus ketika melihat objek yang dekat
- 11 ☒ Sering Berair Jika Membaca Lama
- 12 ☐ Mata lelah atau sakit kepala usai melihat pada jarak dekat dalam waktu lama, misalnya menulis, membaca atau menggunakan Komputer
- 13 ☒ Pandangan kabur seperti berkabut
- 14 ☐ Warna di sekitar terlihat memudar
- 15 ☐ Melihat lingkaran cahaya di sekeliling sumber cahaya
- 16 ☐ Pandangan ganda
- 17 ☐ Penglihatan Lebih Nyaman Dimalam Hari
- 18 ☒ Sering mengganti ukuran kacamata

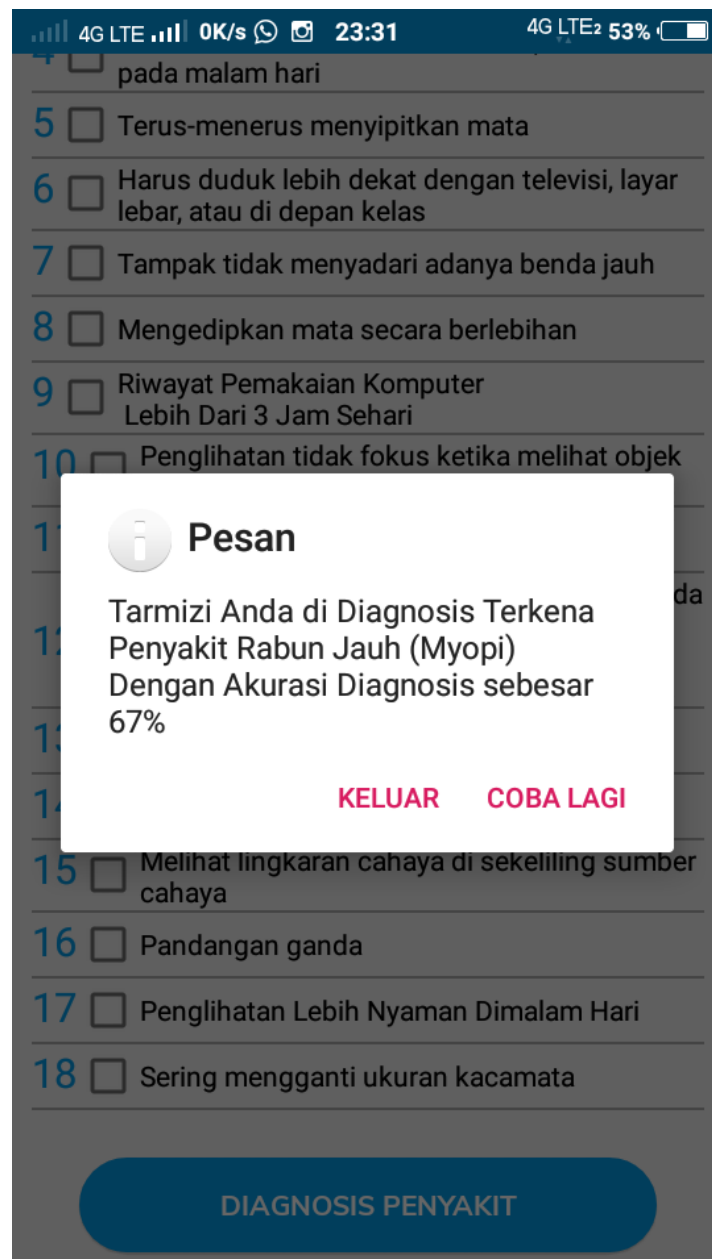
DIAGNOSIS PENYAKIT

Gambar 4.19 Input Data Konsultasi Kasus ke 2 Rabun Jauh

c. Proses Forward Chaining

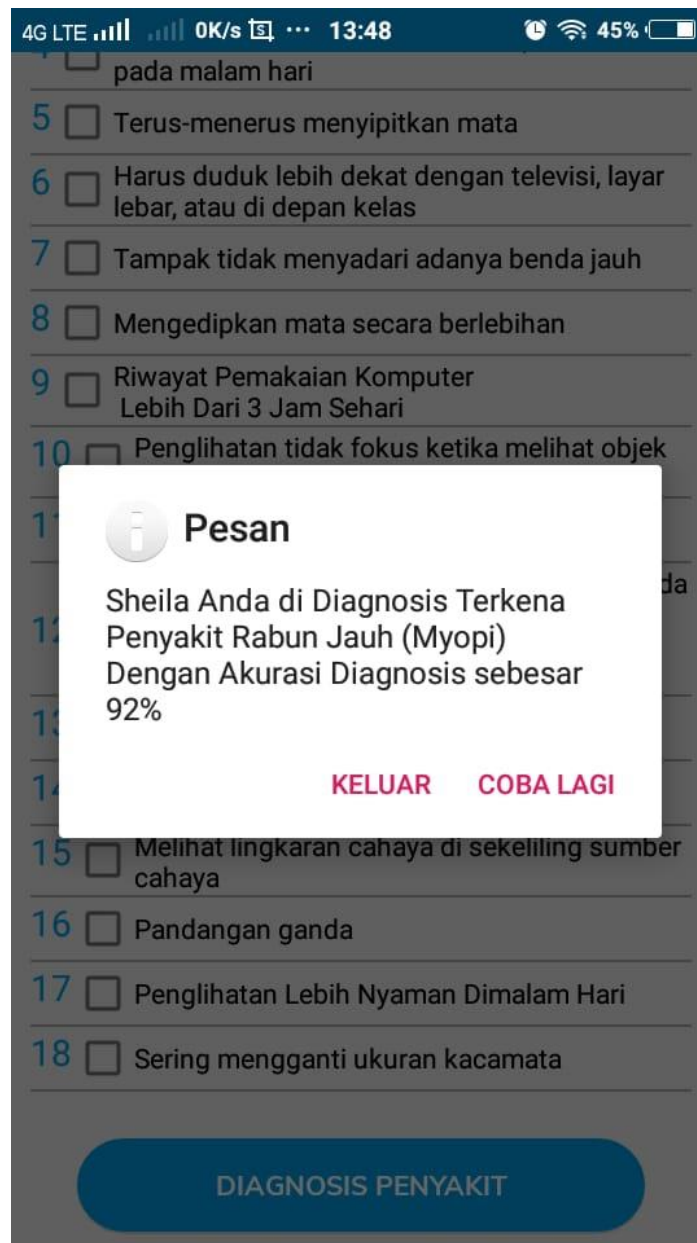
Kemudian perhitungan menggunakan metode *forward chaining* sehingga kita dapat melihat hasil konsultasi yang sudah diinputkan sebelumnya.

Kasus ke -1



Gambar 4.20 Proses *Forward Chaining* Rabun Jauh

Kasus ke -2



Gambar 4.21 Proses *Forward Chaining* Kasus Ke 2 Rabun Jauh

d. Perhitungan Akurasi

Didalam aplikasi diatas *user* mengalami penyakit mata rabun jauh. Berikut perhitungan akurasi untuk rabun jauh yaitu :

Kasus ke 1

$$\text{Akurasi} = \frac{8}{12} \times 100\% = 66,667 \%$$

Kasus ke 2

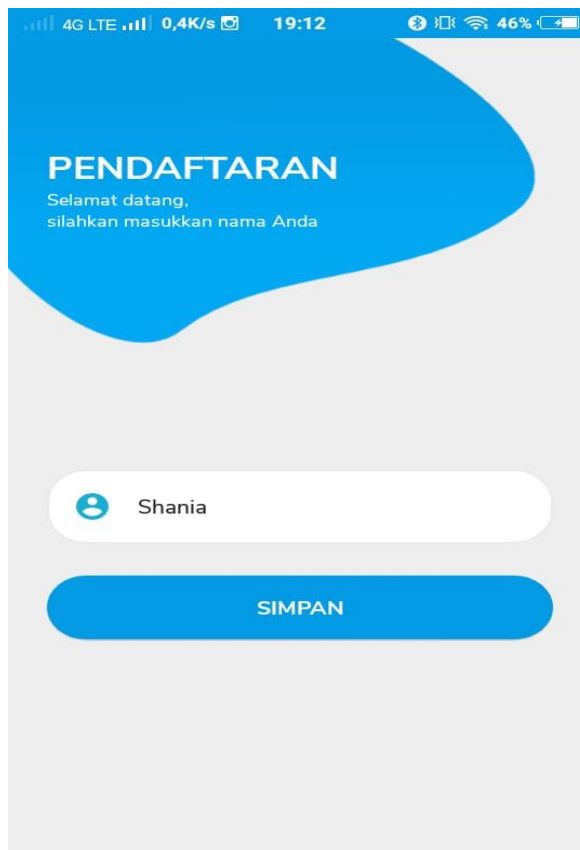
$$\text{Akurasi} = \frac{11}{12} \times 100\% = 91,66 \%$$

2. Mata Rabun Dekat


a. Menginput Data Registrasi

Pertama kita akan menginputkan data registrasi yang diberikan oleh system sebagai berikut :

Kasus ke -1



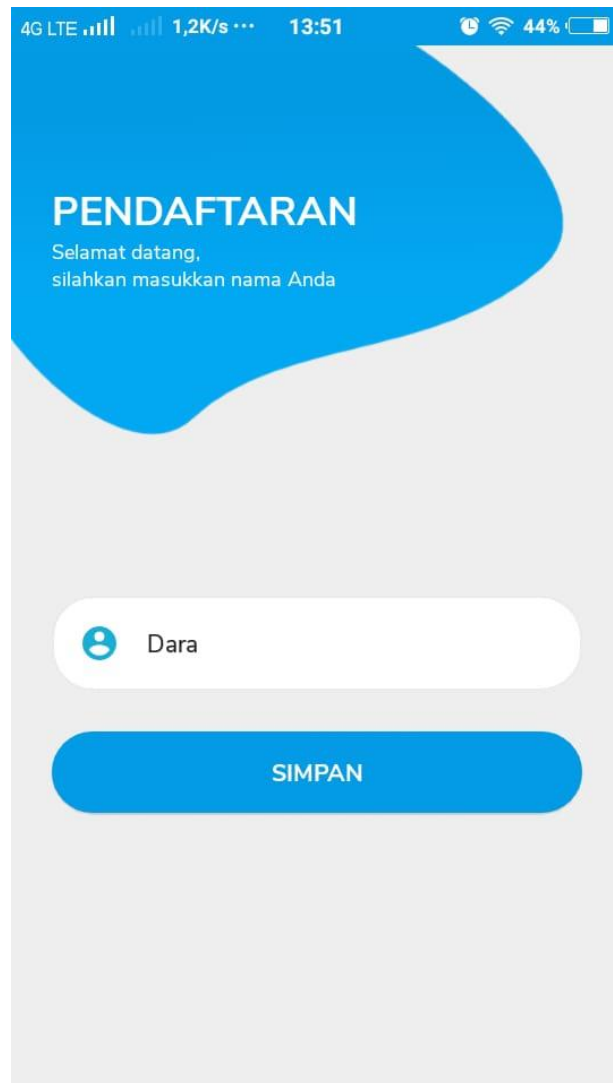
PENDAFTARAN
Selamat datang,
silahkan masukkan nama Anda

 Shania

SIMPAN

Gambar 4.22 Input Data Registrasi Rabun dekat


Kasus ke 2



4G LTE 1,2K/s 13:51 44%

PENDAFTARAN

Selamat datang,
silahkan masukkan nama Anda

 Dara

SIMPAN

Gambar 4.23 Input Data Registrasi Kasus Ke 2 Rabun dekat

b. Menginput Data Diagnosis

Kemudian kita akan menginputkan data konsultasi sesuai yang sudah ditentukan oleh sistem.

Kasus ke 1

DAFTAR DIAGNOSIS
Silahkan Pilih Diagnosis yang anda alami

1	<input type="checkbox"/>	Penglihatan buram ketika melihat benda dalam jarak jauh
2	<input checked="" type="checkbox"/>	Harus menyipitkan mata atau sebagian menutup kelopak mata untuk melihat dengan jelas
3	<input checked="" type="checkbox"/>	Sakit kepala karena kelelahan mata
4	<input checked="" type="checkbox"/>	Kesulitan melihat saat berkendara, terutama pada malam hari
5	<input type="checkbox"/>	Terus-menerus menyipitkan mata
6	<input type="checkbox"/>	Harus duduk lebih dekat dengan televisi, layar lebar, atau di depan kelas
7	<input type="checkbox"/>	Tampak tidak menyadari adanya benda jauh
8	<input checked="" type="checkbox"/>	Mengedipkan mata secara berlebihan
9	<input checked="" type="checkbox"/>	Riwayat Pemakaian Komputer Lebih Dari 3 Jam Sehari
10	<input checked="" type="checkbox"/>	Penglihatan tidak fokus ketika melihat objek yang dekat
5	<input type="checkbox"/>	Terus-menerus menyipitkan mata
6	<input type="checkbox"/>	Harus duduk lebih dekat dengan televisi, layar lebar, atau di depan kelas
7	<input type="checkbox"/>	Tampak tidak menyadari adanya benda jauh
8	<input checked="" type="checkbox"/>	Mengedipkan mata secara berlebihan
9	<input checked="" type="checkbox"/>	Riwayat Pemakaian Komputer Lebih Dari 3 Jam Sehari
10	<input checked="" type="checkbox"/>	Penglihatan tidak fokus ketika melihat objek yang dekat
11	<input checked="" type="checkbox"/>	Sering Berait Jika Membaca Lama
12	<input checked="" type="checkbox"/>	Mata lelah atau sakit kepala usai melihat pada jarak dekat dalam waktu lama, misalnya menulis, membaca atau menggunakan Komputer
13	<input type="checkbox"/>	Pandangan kabur seperti berkabut
14	<input type="checkbox"/>	Warna di sekitar terlihat memudar
15	<input type="checkbox"/>	Melihat lingkaran cahaya di sekeliling sumber cahaya
16	<input type="checkbox"/>	Pandangan ganda
17	<input type="checkbox"/>	Penglihatan Lebih Nyaman Dimalam Hari
18	<input type="checkbox"/>	Sering mengganti ukuran kacamata

DIAGNOSIS PENYAKIT

Gambar 4.24 Input Data Konsultasi Rabun Dekat

Kasus ke 2

DAFTAR DIAGNOSIS
Silahkan Pilih Diagnosis yang anda alami

1	<input type="checkbox"/>	Penglihatan buram ketika melihat benda dalam jarak jauh
2	<input checked="" type="checkbox"/>	Harus menyipitkan mata atau sebagian menutup kelopak mata untuk melihat dengan jelas
3	<input checked="" type="checkbox"/>	Sakit kepala karena kelelahan mata
4	<input checked="" type="checkbox"/>	Kesulitan melihat saat berkendara, terutama pada malam hari
5	<input checked="" type="checkbox"/>	Terus-menerus menyipitkan mata
6	<input type="checkbox"/>	Harus duduk lebih dekat dengan televisi, layar lebar, atau di depan kelas
7	<input type="checkbox"/>	Tampak tidak menyadari adanya benda jauh
8	<input checked="" type="checkbox"/>	Mengedipkan mata secara berlebihan
9	<input checked="" type="checkbox"/>	Riwayat Pemakaian Komputer Lebih Dari 3 Jam Sehari
10	<input checked="" type="checkbox"/>	Penglihatan tidak fokus ketika melihat objek yang dekat
11	<input checked="" type="checkbox"/>	Sering Berait Jika Membaca Lama
12	<input type="checkbox"/>	Mata lelah atau sakit kepala usai melihat pada jarak dekat dalam waktu lama, misalnya menulis, membaca atau menggunakan Komputer
13	<input type="checkbox"/>	Pandangan kabur seperti berkabut
14	<input type="checkbox"/>	Warna di sekitar terlihat memudar
15	<input type="checkbox"/>	Melihat lingkaran cahaya di sekeliling sumber cahaya
16	<input type="checkbox"/>	Pandangan ganda
17	<input type="checkbox"/>	Penglihatan Lebih Nyaman Dimalam Hari
18	<input checked="" type="checkbox"/>	Sering mengganti ukuran kacamata

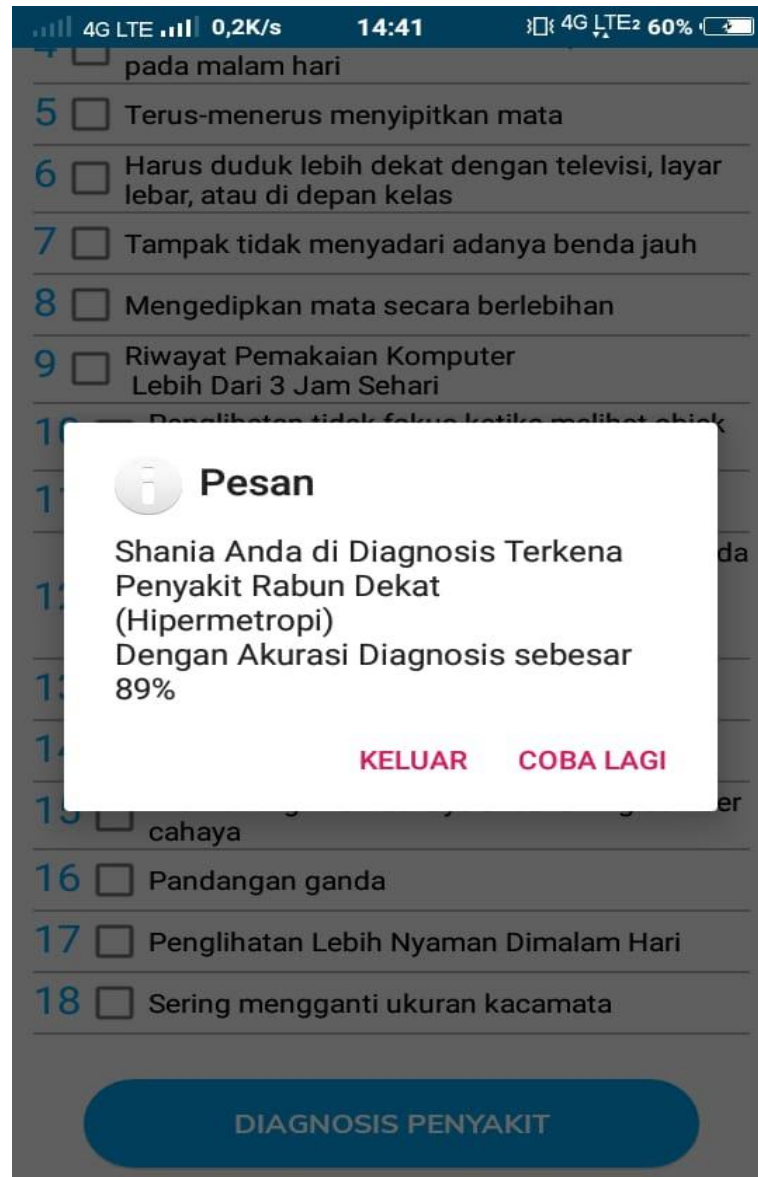
DIAGNOSIS PENYAKIT

Gambar 4.25 Input Data Konsultasi Kasus ke 2 Rabun Dekat

c. Proses Forward Chaining

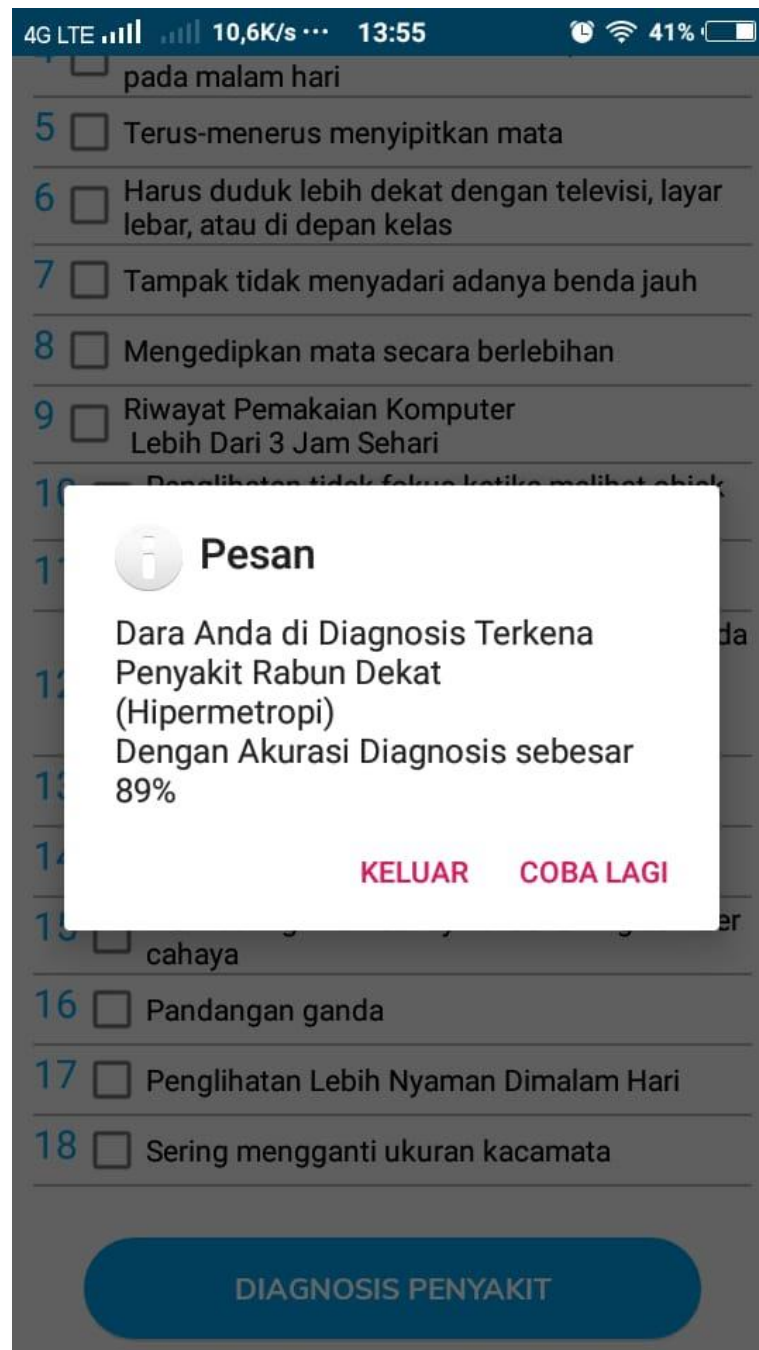
Kemudian perhitungan menggunakan metode *forward chaining* sehingga kita dapat melihat hasil konsultasi yang sudah diinputkan sebelumnya.

Kasus ke 1



Gambar 4.26 Proses *Forward Chaining* Rabun Dekat

Kasus ke 2



Gambar 4.27 Proses *Forward Chaining* Kasus ke 2 Rabun Dekat

d. Perhitungan Akurasi

Didalam aplikasi diatas *Shania* mengalami penyakit mata rabun dekat. Berikut perhitungan akurasi untuk rabun dekat yaitu :

Kasus Ke 1

$$\text{Akurasi} = \frac{8}{9} \times 100\% = 88,88 \%$$

Kasus ke 2

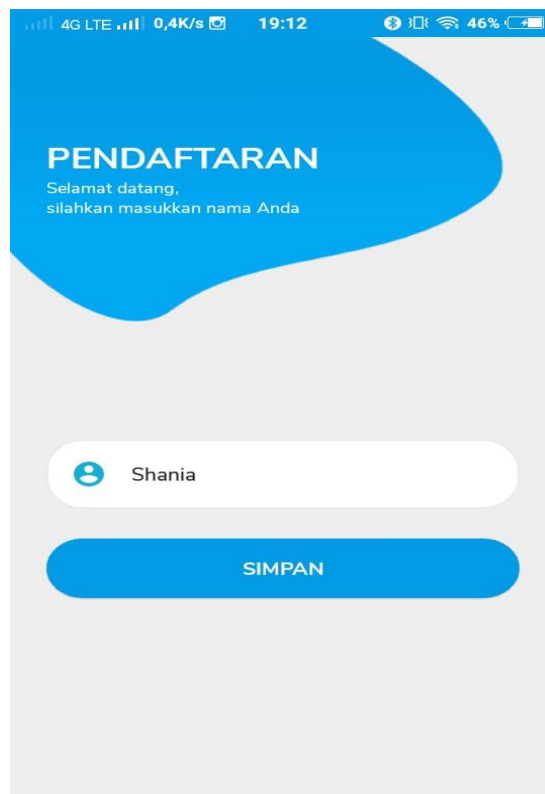
$$\text{Akurasi} = \frac{8}{9} \times 100\% = 88,88 \%$$

3. Mata Katarak Senilis


a. Menginput Data Registrasi

Pertama kita akan menginputkan data registrasi yang diberikan oleh system sebagai berikut :

Kasus ke -1



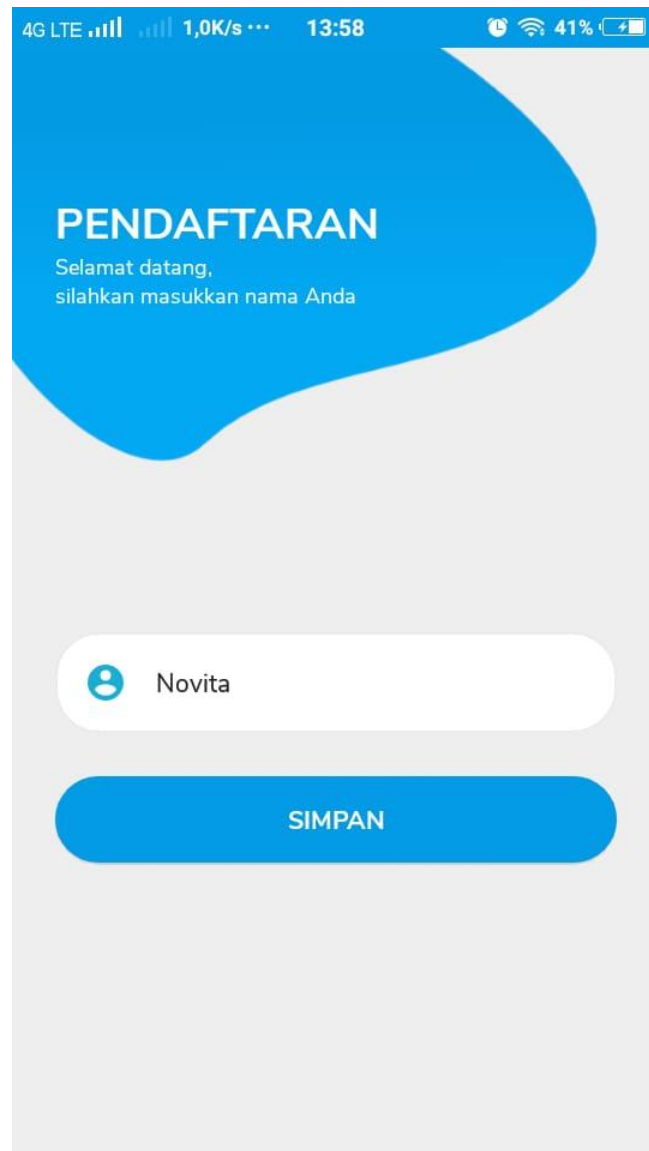
PENDAFTARAN
Selamat datang,
silahkan masukkan nama Anda

 Shania

SIMPAN

Gambar 4.28 Input Data Registrasi Katarak Senilis

Kasus ke -2



The screenshot shows a mobile application interface for registration. At the top, the status bar displays '4G LTE', signal strength, '1,0K/s', the time '13:58', and a 41% battery level. The main header area is blue with the word 'PENDAFTARAN' in white. Below this, a message reads 'Selamat datang, silahkan masukkan nama Anda'. A white input field contains the name 'Novita' next to a person icon. Below the input field is a blue button labeled 'SIMPAN'.

Gambar 4.29 Input Data Registrasi Kasus Ke 2 Katarak Senilis

b. Menginput Data Diagnosis

Kemudian kita akan menginputkan data konsultasi sesuai yang sudah di tentukan oleh sistem.

Kasus ke -1

DAFTAR DIAGNOSIS
Silahkan Pilih Diagnosis yang anda alami

1 <input type="checkbox"/>	Penglihatan buram ketika melihat benda dalam jarak jauh	5 <input type="checkbox"/>	Terus-menerus menyipitkan mata
2 <input type="checkbox"/>	Harus menyipitkan mata atau sebagian menutup kelopak mata untuk melihat dengan jelas	6 <input type="checkbox"/>	Harus duduk lebih dekat dengan televisi, layar lebar, atau di depan kelas
3 <input type="checkbox"/>	Sakit kepala karena kelelahan mata	7 <input type="checkbox"/>	Tampak tidak menyadari adanya benda jauh
4 <input checked="" type="checkbox"/>	Kesulitan melihat saat berkendara, terutama pada malam hari	8 <input checked="" type="checkbox"/>	Mengedipkan mata secara berlebihan
5 <input type="checkbox"/>	Terus-menerus menyipitkan mata	9 <input type="checkbox"/>	Riwayat Pemakaian Komputer Lebih Dari 3 Jam Sehari
6 <input type="checkbox"/>	Harus duduk lebih dekat dengan televisi, layar lebar, atau di depan kelas	10 <input type="checkbox"/>	Penglihatan tidak fokus ketika melihat objek yang dekat
7 <input type="checkbox"/>	Tampak tidak menyadari adanya benda jauh	11 <input type="checkbox"/>	Sering Berait Jika Membaca Lama
8 <input type="checkbox"/>	Mengedipkan mata secara berlebihan	12 <input type="checkbox"/>	Mata lelah atau sakit kepala usai melihat pada jarak dekat dalam waktu lama, misalnya menulis, membaca atau menggunakan Komputer
9 <input type="checkbox"/>	Riwayat Pemakaian Komputer Lebih Dari 3 Jam Sehari	13 <input checked="" type="checkbox"/>	Pandangan kabur seperti berkabut
10 <input type="checkbox"/>	Penglihatan tidak fokus ketika melihat objek yang dekat	14 <input checked="" type="checkbox"/>	Warna di sekitar terlihat memudar
		15 <input checked="" type="checkbox"/>	Melihat lingkaran cahaya di sekeliling sumber cahaya
		16 <input checked="" type="checkbox"/>	Pandangan ganda
		17 <input checked="" type="checkbox"/>	Penglihatan Lebih Nyaman Dimalam Hari
		18 <input checked="" type="checkbox"/>	Sering mengganti ukuran kacamata

DIAGNOSIS PENYAKIT

Gambar 4.30 Input Data Konsultasi Katarak Senilis

Kasus ke 2

DAFTAR DIAGNOSIS
Silahkan Pilih Diagnosis yang anda alami

1 <input checked="" type="checkbox"/> Penglihatan buram ketika melihat benda dalam jarak jauh	5 <input type="checkbox"/> Terus-menerus menyipitkan mata
2 <input checked="" type="checkbox"/> Harus menyipitkan mata atau sebagian menutup kelopak mata untuk melihat dengan jelas	6 <input type="checkbox"/> Harus duduk lebih dekat dengan televisi, layar lebar, atau di depan kelas
3 <input type="checkbox"/> Sakit kepala karena kelelahan mata	7 <input type="checkbox"/> Tampak tidak menyadari adanya benda jauh
4 <input checked="" type="checkbox"/> Kesulitan melihat saat berkendara, terutama pada malam hari	8 <input type="checkbox"/> Mengedipkan mata secara berlebihan
5 <input type="checkbox"/> Terus-menerus menyipitkan mata	9 <input type="checkbox"/> Riwayat Pemakaian Komputer Lebih Dari 3 Jam Sehari
6 <input type="checkbox"/> Harus duduk lebih dekat dengan televisi, layar lebar, atau di depan kelas	10 <input type="checkbox"/> Penglihatan tidak fokus ketika melihat objek yang dekat
7 <input type="checkbox"/> Tampak tidak menyadari adanya benda jauh	11 <input checked="" type="checkbox"/> Sering Berair Jika Membaca Lama
8 <input type="checkbox"/> Mengedipkan mata secara berlebihan	12 <input type="checkbox"/> Mata lelah atau sakit kepala usai melihat pada jarak dekat dalam waktu lama, misalnya menulis, membaca atau menggunakan Komputer
9 <input type="checkbox"/> Riwayat Pemakaian Komputer Lebih Dari 3 Jam Sehari	13 <input checked="" type="checkbox"/> Pandangan kabur seperti berkabut
10 <input type="checkbox"/> Penglihatan tidak fokus ketika melihat objek yang dekat	14 <input checked="" type="checkbox"/> Warna di sekitar terlihat memudar
	15 <input checked="" type="checkbox"/> Melihat lingkaran cahaya di sekeliling sumber cahaya
	16 <input checked="" type="checkbox"/> Pandangan ganda
	17 <input checked="" type="checkbox"/> Penglihatan Lebih Nyaman Dimalam Hari
	18 <input type="checkbox"/> Sering mengganti ukuran kacamata

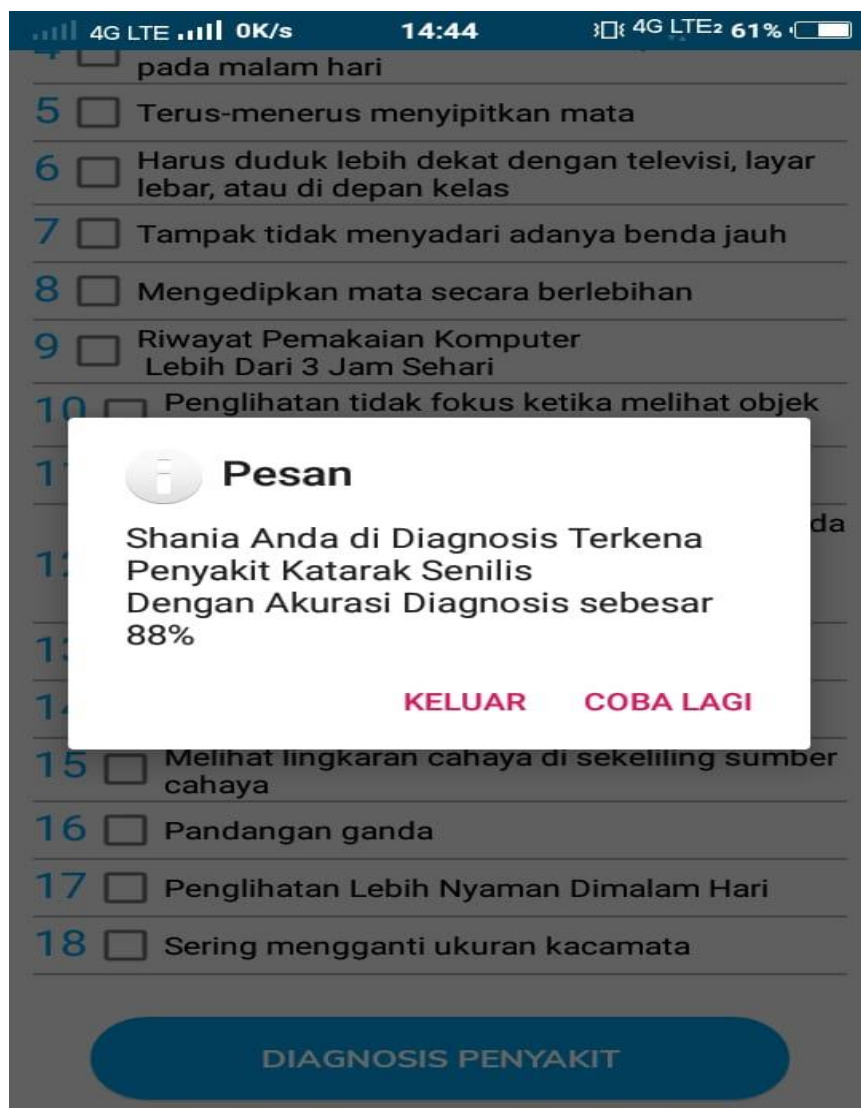
DIAGNOSIS PENYAKIT

Gambar 4.31 Input Data Konsultasi Kasus ke 2 Katarak Senilis

c. Proses Forward Chaining

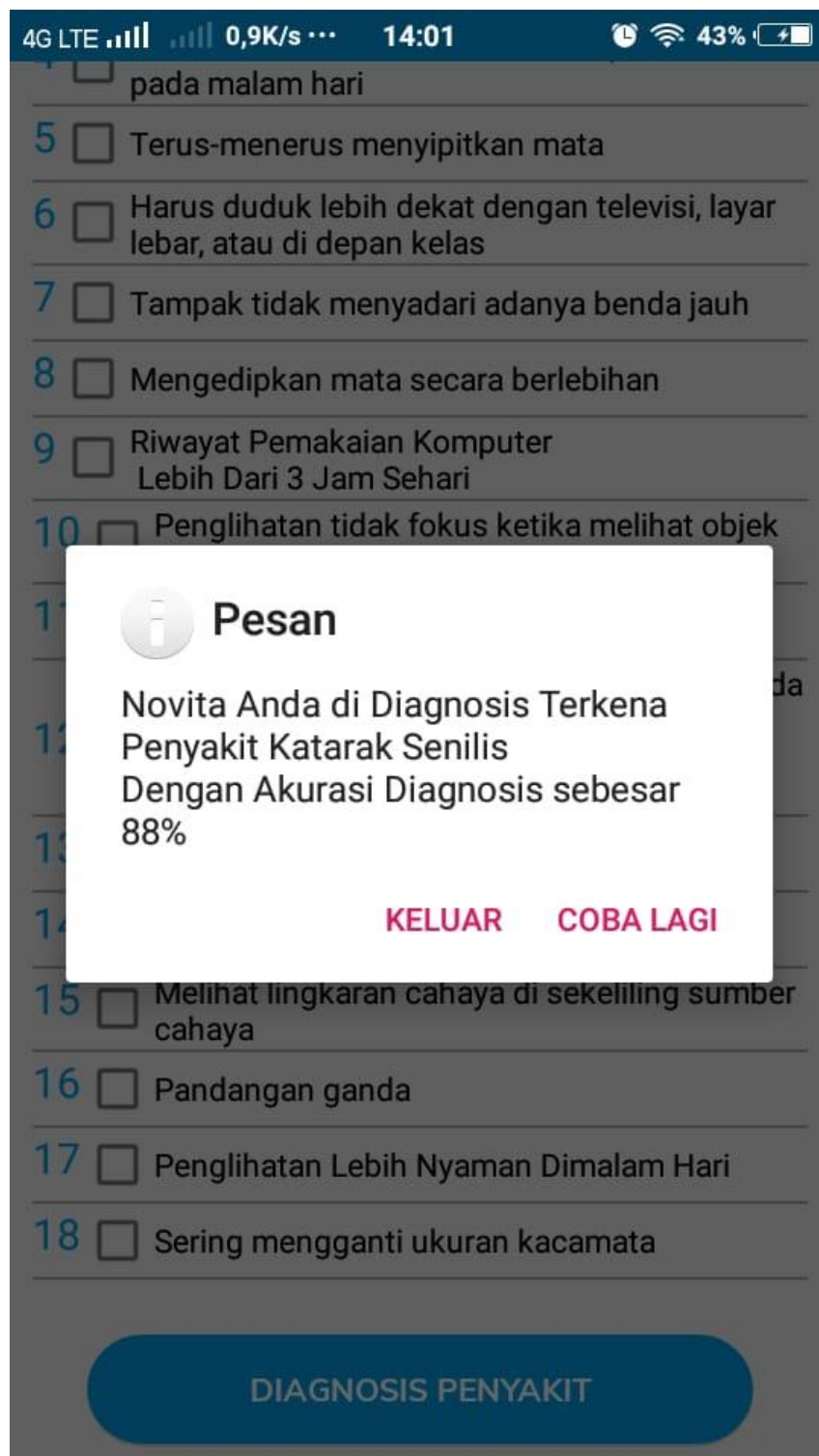
Kemudian perhitungan menggunakan metode *forward chaining* sehingga kita dapat melihat hasil konsultasi yang sudah diinputkan sebelumnya.

Kasus ke 1



Gambar 4.32 Proses *Forward Chaining* Katarak Senilis

Kasus ke -2



Gambar 4.33 Proses *Forward Chaining* Katarak Senilis

d. Perhitungan Akurasi

Didalam aplikasi diatas *user* mengalami penyakit mata katarak senilis. Berikut perhitungan akurasi untuk katarak senilis yaitu :

Kasus ke -1

$$\text{Akurasi} = \frac{7}{8} \times 100\% = 87,5 \%$$

Kasus ke 2

$$\text{Akurasi} = \frac{7}{8} \times 100\% = 87,5 \%$$

e. Forward Chaining Source Code

```
public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
```

```
    if (buttonView.getId() == R.id.G01) {
        if (isChecked) {
            daftarPenyakit.add("G01");
        } else {
            daftarPenyakit.remove("G01");
        }

    } else if (buttonView.getId() == R.id.G02) {
        if (isChecked) {
            daftarPenyakit.add("G02");
        } else {
            daftarPenyakit.remove("G02");
        }

    } else if (buttonView.getId() == R.id.G03) {
        if (isChecked) {
            daftarPenyakit.add("G03");
```

```
    } else {  
        daftarPenyakit.remove("G03");  
    }  
  
} else if (buttonView.getId() == R.id.G04) {  
    if (isChecked) {  
        daftarPenyakit.add("G04");  
    } else {  
        daftarPenyakit.remove("G04");  
    }  
  
} else if (buttonView.getId() == R.id.G05) {  
    if (isChecked) {  
        daftarPenyakit.add("G05");  
    } else {  
        daftarPenyakit.remove("G05");  
    }  
  
} else if (buttonView.getId() == R.id.G06) {  
    if (isChecked) {  
        daftarPenyakit.add("G06");  
    } else {  
        daftarPenyakit.remove("G06");  
    }  
  
} else if (buttonView.getId() == R.id.G07) {  
    if (isChecked) {  
        daftarPenyakit.add("G07");  
    } else {  
        daftarPenyakit.remove("G07");  
    }  
}
```



```

} else if (buttonView.getId() == R.id.G08) {
    if (isChecked) {
        daftarPenyakit.add("G08");
    } else {
        daftarPenyakit.remove("G08");
    }

} else if (buttonView.getId() == R.id.G09) {
    if (isChecked) {
        daftarPenyakit.add("G09");
    } else {
        daftarPenyakit.remove("G09");
    }

} else if (buttonView.getId() == R.id.G10) {
    if (isChecked) {
        daftarPenyakit.add("G10");
    } else {
        daftarPenyakit.remove("G10");
    }

} else if (buttonView.getId() == R.id.G11) {
    if (isChecked) {
        daftarPenyakit.add("G11");
    } else {
        daftarPenyakit.remove("G11");
    }

} else if (buttonView.getId() == R.id.G12) {
    if (isChecked) {

```

```

        daftarPenyakit.add("G12");
    } else {
        daftarPenyakit.remove("G12");
    }

} else if (buttonView.getId() == R.id.G13) {
    if (isChecked) {
        daftarPenyakit.add("G13");
    } else {
        daftarPenyakit.remove("G13");
    }
}

} else if (buttonView.getId() == R.id.G14) {
    if (isChecked) {
        daftarPenyakit.add("G14");
    } else {
        daftarPenyakit.remove("G14");
    }
}

} else if (buttonView.getId() == R.id.G15) {
    if (isChecked) {
        daftarPenyakit.add("G15");
    } else {
        daftarPenyakit.remove("G15");
    }
}

} else if (buttonView.getId() == R.id.G16) {
    if (isChecked) {
        daftarPenyakit.add("G16");
    } else {
        daftarPenyakit.remove("G16");
    }
}

```

```

    }

    } else if (buttonView.getId() == R.id.G17) {
        if (isChecked) {
            daftarPenyakit.add("G17");
        } else {
            daftarPenyakit.remove("G17");
        }
    }

    } else if (buttonView.getId() == R.id.G18) {
        if (isChecked) {
            daftarPenyakit.add("G18");
        } else {
            daftarPenyakit.remove("G18");
        }
    }
}

```

4.2.2 Penerapan

Implementasi dan pengujian sistem dilakukan untuk memastikan rangkaian yang dihasilkan berfungsi sesuai rencana atau tidak. Setiap pengujian dilakukan melalui pengukuran yang nantinya akan digunakan untuk menganalisis perangkat keras dan lunak pendukung (Mhd Furqan, 2020). Menggunakan sistem pakar dan menerapkan metode rantai langsung, yang dirancang untuk menyederhanakan tugas pasien mengetahui jenis penyakit mata rabun. Dan sebagai pengetahuan bagi para pengguna internet tentang pentingnya menjaga kesehatan mata dari segi gejala mata miopia pada sistem. Penerapan sistem ini akan diterapkan pada pengguna yang menderita miopia. Sistem ini akan memudahkan pengguna untuk secara konsisten dan cepat mengidentifikasi penyakit mata rabun dibandingkan dengan cara manual.

Penerapan memudahkan anda untuk menentukan jenis penyakit mata rabun berdasarkan gejala yang dia rasakan. Menggunakan sistem pakar dan menerapkan metode rantai langsung yang dirancang untuk memudahkan pasien dalam mengidentifikasi jenis penyakit mata rabun. Dan sebagai pengetahuan bagi para pengguna internet tentang pentingnya menjaga kesehatan mata dari segi gejala mata miopia pada sistem. Pengujian akan dimulai dari penjelasan tentang program kepada orang tersebut dan mereka akan diminta menginput data dan mengisi pertanyaan yang diberikan oleh sistem dengan gejala mereka rasakan. Langkah pertama user membuka aplikasi, kemudian melakukan registrasi biodata diri untuk masuk ke dalam sistem lalu klik simpan. Secara otomatis data tersebut tersimpan kedalam data base. Selanjutnya user menjawab beberapa pertanyaan di dalam aplikasi dengan menjawab True and False. Setelah user menjawab pertanyaan tersebut, user mengklik tombol lanjutkan. Laporan hasil jawaban yang telah diisi oleh user akan terlihat secara otomatis. kemudian klik tombol keluar untuk keluar dari program.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang dilakukan dalam pembuatan sistem ini maka dapat diambil kesimpulan sebagai berikut:

- 1 Aplikasi ini dapat mendiagnosis dan mengetahui hasil diagnosa penyakit mata rabun pada pasien dengan menggunakan metode rantai langsung yaitu metode rantai langsung. berlaku untuk aplikasi dalam bentuk membangun hubungan aturan dengan memilih gejala dan solusi. Gejala dan solusi dimasukkan satu per satu sesuai dengan gejalanya.
- 2 Sistem aplikasi *mobile* dapat digunakan pengguna dalam mendiagnosis gejala dan mendapatkan pengetahuan terkait mata rabun, Aplikasi ini dapat menemukan solusi untuk jenis penyakit mata rabun sesuai dengan gejala yang dialami pasien.

2.1 Saran

Saran untuk pengembangan aplikasi kedepannya sebagai berikut :

- 1 Sebaiknya sistem aplikasi untuk mata rabun ini dapat dikembangkan dari segi gejala, penyakit dan solusinya sehingga dapat memberikan informasi yang cepat dan lebih berharga.
- 2 Diharapkan sistem ini dapat diupdate kembali terutama dari segi tampilan agar lebih menarik dan mudah digunakan.

DAFTAR PUSTAKA

- Bagus Fery, Indah Werdiningsih dan Endah Purwanti 2017, *Aplikasi Sistem Pakar Diagnosa Penyakit Pada Anak Bawah Lima Tahun Menggunakan Metode Forward Chaining*. Journal of Information Systems Engineering and Business Intelligence Vol. 3, No. 1.
- Evi Dewi dan Irna Nur Restianie 2016, *Aplikasi Sistem Pakar Untuk Mendiagnosis Penyakit Anak (Balita) Dengan Menggunakan Metode Forward Chaining*. Jurnal STMIK AMIKOM Yogyakarta.
- Taufik Ramadhan dan Victor G Utomo 2014, *Rancang Bangun Aplikasi Mobile Untuk Notifikasi Jadwal Kuliah Berbasis Android (Studi Kasus : Stmik Provisi Semarang)*. Program Studi Teknik Informatika STMIK PROVISI Semarang.
- Tatag Abiyoso Utomo, Bambang Darmo Yuwono, dan Fauzi Janu Amarrohman 2017, *Aplikasi Sistem Informasi Geografis Berbasis Web Dan Android Untuk Pemilihan Jalur Alternatif Menuju Tempat Pariwisata (Studi Kasus: Kota Wisata Cibubur Dan Jungleland, Kabupaten Bogor)*. Jurnal Geodesi UNDIP.
- Hasanah, Ridarmin, dan Sukri Adrianto 2017, *Aplikasi Sistem Pakar Pendeteksi Kerusakan Laptop/Pc Dengan Penerapan Metode Forward Chaining Menggunakan Bahasa Pemrograman JAVA*. Jurnal Informatika, Manajemen dan Komputer, Vol. 9 No. 2 , Desember 2017.
- Windah Supartini dan Hindarto 2016, *Sistem Pakar Berbasis Web Dengan Metode Forward Chaining Dalam Mendiagnosis Dini Penyakit Tuberkulosis di Jawa Timur*. KINETIK, Vol.1, No.3, 2016, November Hal. 147-154.

Yunia Ervinaeni, Aziz Setyawan Hidayat dan Eri Riana 2019, *Sistem Pakar Diagnosa Gangguan Hiperaktif Pada Anak Dengan Metode Naive Bayes Berbasis Web*, JURNAL MEDIA INFORMATIKA BUDIDARMA, Vol 3, No 2, April 2019.

Yusuf Ramadhan Nasution 2018, *Penerapan Aplikasi Online Angket Persepsi Mahasiswa Terhadap Kinerja Dosen Uin Sumatera Utara Medan*, JISTech, Vol.3, No.2, Juli - Desember 2018.

Abdullah Husin, Usman dan Muhammad Putra Faren 2018, *Sistem Pakar Pendeteksi Penyakit Berdasarkan Keluhan Buang Air Kecil Menggunakan Metode Forward Chaining*, JURNAL IPTEK TERAPAN.

Iqbal Kamil Siregar dan Faisal Taufik 2017, *Perancangan Aplikasi Sms Alert Berbasis Web*, JIMP - Jurnal Informatika Merdeka Pasuruan Vol.2 No.2 Agustus 2017.

Haspiani M, *Karakteristik Penderita Katarak Senilis Yang Telah Dilakukan Pembedahan Katarak Di Rumah Sakit Pendidikan Universitas Hasanuddin Periode 1 Januari 2017 – 30 Juni 2017*, Skripsi Desember 2017 Fakultas Kedokteran Universitas Hasanuddin Makassar.

Mhd Furqan, Rakhmat Kurniawan, Indri Gusmita Br Rambe 2020, *Tempat Sampah Pintar Dengan Logika Fuzzy Berbasis NodeMCU*. Indonesian Journal of Computer Science ISSN 2302-4364 (print) dan 2549-7286 (online)

Yusuf Ramadhan Nasution Dan Khairuna 2017, *Sistem Pakar Deteksi Awal Penyakit Tuberkulosis Dengan Metode Bayes*, KLOROFIL Vol. 1 No. 1, 2017: 17-23.

LAMPIRAN - LAMPIRAN

Lampiran 1

```
package com.hafiz.forwardchaining.activity;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import com.hafiz.forwardchaining.R;

public class RegisterActivity extends AppCompatActivity implements
View.OnClickListener {

    private EditText et_name_regigster;
    private Button btnSave;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);
        initView();
        initIntitalize();
    }

    private void initIntitalize() {
        btnSave.setOnClickListener(this);
    }

    private void initView() {
        et_name_regigster = findViewById(R.id.et_name_regigster);
        btnSave = findViewById(R.id.btn_save);
    }

    private boolean validation(String name){
```



```

        if (!name.isEmpty()){
            Intent mainMenu = new Intent(this,MainMenu.class);
            mainMenu.putExtra("name",name);
            mainMenu.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK |
Intent.FLAG_ACTIVITY_NEW_TASK);
            startActivity(mainMenu);
            this.finish();
        }else{
            Toast.makeText(this,"Nama Tidak Boleh
Kosong",Toast.LENGTH_SHORT).show();
        }
        return false;
    }

    @Override
    public void onClick(View v) {
        if (v.getId()==R.id.btn_save){
            validation(et_name_regigster.getText().toString());
        }
    }
}

```

Main Menu Activity

```

package com.hafiz.forwardchaining.activity;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.TextView;

```

```
import com.hafiz.forwardchaining.R;
```

```
public class MainMenu extends AppCompatActivity implements  
View.OnClickListener {
```

```
    private TextView tv_auth_subtitle;
```

```
    private Button btnSolusi,btnKonsultasi;
```

```
    private ImageView btnLogout;
```

```
    private String name;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main_menu);
```

```
    initView();
```

```
    initInitialize();
```

```
}
```

```
private void initInitialize() {
```

```
Intent intent = getIntent();
```

```
if (intent!=null){
```

```
    name = intent.getStringExtra("name");
```

```
    tv_auth_subtitle.setText(String.format("Selamat Datang %s", name));
```

```
}
```

```
    btnSolusi.setOnClickListener(this);
```

```
    btnKonsultasi.setOnClickListener(this);
```

```
    btnLogout.setOnClickListener(this);
```

```
}
```

```
private void initView() {
```

```
    tv_auth_subtitle = findViewById(R.id.tv_auth_subtitle);
```

```
    btnSolusi = findViewById(R.id.btnsolusi);
```

```
    btnKonsultasi = findViewById(R.id.btnConsult);
```

```
    btnLogout = findViewById(R.id.btnLogout);
```

```
}
```

```
@Override
```

```
public void onClick(View v) {
```

```

if (v.getId()==R.id.btnsolusi){

    Intent mainMenu = new Intent(this,SolutionActivity.class);

    startActivity(mainMenu);

} else if (v.getId()==R.id.btnConsult){

    Intent mainMenu = new Intent(this, ConsultActivity.class);

    mainMenu.putExtra("name",name);

    startActivity(mainMenu);

} else if (v.getId()==R.id.btnLogout){

    Intent register = new Intent(this,RegisterActivity.class);

    register.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);

    startActivity(register);

    finish();

}

}

}

```

Consult Activity

```

package com.hafiz.forwardchaining.activity;

```

```

import android.annotation.SuppressWarnings;

```

```

import android.content.Intent;

```

```
import android.os.Build;

import android.os.Bundle;

import android.os.Handler;

import android.view.View;

import android.widget.Button;

import android.widget.CheckBox;

import android.widget.CompoundButton;

import android.widget.ImageView;

import androidx.annotation.RequiresApi;

import com.hafiz.forwardchaining.BaseActivity;

import com.hafiz.forwardchaining.R;

import com.hafiz.forwardchaining.models.Sortbyroll;

import com.hafiz.forwardchaining.models.penyakit;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Collections;

import java.util.Comparator;

import java.util.Iterator;
```

```

public class ConsultActivity extends BaseActivity implements
CompoundButton.OnCheckedChangeListener, View.OnClickListener {

    private CheckBox G01, G02, G03, G04, G05, G06, G07, G08, G09, G10,
G11, G12, G13, G14, G15, G16, G17, G18;

    private Button btnDiagnosis;

    private String name;

    private ArrayList<String> daftarPenyakit = new ArrayList<>();

    private ArrayList<Object> A = new ArrayList<>();

    private ArrayList<penyakit> rangkum = new ArrayList<>();

    private ArrayList<String> P01 = new ArrayList<>(Arrays.asList("G01",
"G02", "G03", "G04", "G05"

    , "G06", "G07", "G08", "G09", "G13", "G17", "G18"));

    private ArrayList<String> P02 = new ArrayList<>(Arrays.asList("G02",
"G03", "G04", "G05"

    , "G08", "G09", "G10", "G11", "G12"));

    private ArrayList<String> P03 = new ArrayList<>(Arrays.asList("G01",
"G04", "G13"

    , "G14", "G15", "G16", "G17", "G18"));

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

```

```
setContentView(R.layout.activity_consult);

initView();

initInitialize();

}

private void initInitialize() {

    Intent intent = getIntent();

    if (intent!=null){

        name = intent.getStringExtra("name");

    }

    ImageView btnBack = findViewById(R.id.btnBack);

    btnBack.setOnClickListener(v -> onBackPressed());

    G01.setOnCheckedChangeListener(this);

    G02.setOnCheckedChangeListener(this);

    G03.setOnCheckedChangeListener(this);

    G04.setOnCheckedChangeListener(this);

    G05.setOnCheckedChangeListener(this);

    G06.setOnCheckedChangeListener(this);

    G07.setOnCheckedChangeListener(this);

    G08.setOnCheckedChangeListener(this);

    G09.setOnCheckedChangeListener(this);
```

```
G10.setOnCheckedChangeListener(this);

G11.setOnCheckedChangeListener(this);

G12.setOnCheckedChangeListener(this);

G13.setOnCheckedChangeListener(this);

G14.setOnCheckedChangeListener(this);

G15.setOnCheckedChangeListener(this);

G16.setOnCheckedChangeListener(this);

G17.setOnCheckedChangeListener(this);

G18.setOnCheckedChangeListener(this);

btnDiagnosis.setOnClickListener(this);

}

private void initView() {

    btnDiagnosis = findViewById(R.id.btnDiagnosis);

    G01 = findViewById(R.id.G01);

    G02 = findViewById(R.id.G02);

    G03 = findViewById(R.id.G03);

    G04 = findViewById(R.id.G04);

    G05 = findViewById(R.id.G05);

    G06 = findViewById(R.id.G06);

    G07 = findViewById(R.id.G07);
```



```
G08 = findViewById(R.id.G08);

G09 = findViewById(R.id.G09);

G10 = findViewById(R.id.G10);

G11 = findViewById(R.id.G11);

G12 = findViewById(R.id.G12);

G13 = findViewById(R.id.G13);

G14 = findViewById(R.id.G14);

G15 = findViewById(R.id.G15);

G16 = findViewById(R.id.G16);

G17 = findViewById(R.id.G17);

G18 = findViewById(R.id.G18);

}
```

@Override

```
public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
```

```
    if (buttonView.getId() == R.id.G01) {
```

```
        if (isChecked) {
```

```
            daftarPenyakit.add("G01");
```

```
        } else {
```

```
            daftarPenyakit.remove("G01");
```

```
    }  
  
    } else if (buttonView.getId() == R.id.G02) {  
  
        if (isChecked) {  
  
            daftarPenyakit.add("G02");  
  
        } else {  
  
            daftarPenyakit.remove("G02");  
  
        }  
  
    } else if (buttonView.getId() == R.id.G03) {  
  
        if (isChecked) {  
  
            daftarPenyakit.add("G03");  
  
        } else {  
  
            daftarPenyakit.remove("G03");  
  
        }  
  
    } else if (buttonView.getId() == R.id.G04) {  
  
        if (isChecked) {  
  
            daftarPenyakit.add("G04");  
  
        } else {  
  
            daftarPenyakit.remove("G04");  
  
        }  
  
    } else if (buttonView.getId() == R.id.G05) {
```

```
    if (isChecked) {

        daftarPenyakit.add("G05");

    } else {

        daftarPenyakit.remove("G05");

    }

} else if (buttonView.getId() == R.id.G06) {

    if (isChecked) {

        daftarPenyakit.add("G06");

    } else {

        daftarPenyakit.remove("G06");

    }

} else if (buttonView.getId() == R.id.G07) {

    if (isChecked) {

        daftarPenyakit.add("G07");

    } else {

        daftarPenyakit.remove("G07");

    }

} else if (buttonView.getId() == R.id.G08) {

    if (isChecked) {

        daftarPenyakit.add("G08");

    }
```

```
    } else {  
  
        daftarPenyakit.remove("G08");  
  
    }  
  
} else if (buttonView.getId() == R.id.G09) {  
  
    if (isChecked) {  
  
        daftarPenyakit.add("G09");  
  
    } else {  
  
        daftarPenyakit.remove("G09");  
  
    }  
  
} else if (buttonView.getId() == R.id.G10) {  
  
    if (isChecked) {  
  
        daftarPenyakit.add("G10");  
  
    } else {  
  
        daftarPenyakit.remove("G10");  
  
    }  
  
} else if (buttonView.getId() == R.id.G11) {  
  
    if (isChecked) {  
  
        daftarPenyakit.add("G11");  
  
    } else {  
  
        daftarPenyakit.remove("G11");  
  
    }  
  
}
```

```
}
```

```
} else if (buttonView.getId() == R.id.G12) {
```

```
    if (isChecked) {
```

```
        daftarPenyakit.add("G12");
```

```
    } else {
```

```
        daftarPenyakit.remove("G12");
```

```
    }
```

```
} else if (buttonView.getId() == R.id.G13) {
```

```
    if (isChecked) {
```

```
        daftarPenyakit.add("G13");
```

```
    } else {
```

```
        daftarPenyakit.remove("G13");
```

```
    }
```

```
} else if (buttonView.getId() == R.id.G14) {
```

```
    if (isChecked) {
```

```
        daftarPenyakit.add("G14");
```

```
    } else {
```

```
        daftarPenyakit.remove("G14");
```

```
    }
```

```
} else if (buttonView.getId() == R.id.G15) {  
  
    if (isChecked) {  
  
        daftarPenyakit.add("G15");  
  
    } else {  
  
        daftarPenyakit.remove("G15");  
  
    }  
  
} else if (buttonView.getId() == R.id.G16) {  
  
    if (isChecked) {  
  
        daftarPenyakit.add("G16");  
  
    } else {  
  
        daftarPenyakit.remove("G16");  
  
    }  
  
} else if (buttonView.getId() == R.id.G17) {  
  
    if (isChecked) {  
  
        daftarPenyakit.add("G17");  
  
    } else {  
  
        daftarPenyakit.remove("G17");  
  
    }  
  
} else if (buttonView.getId() == R.id.G18) {  
  
    if (isChecked) {
```

```

        daftarPenyakit.add("G18");

    } else {

        daftarPenyakit.remove("G18");

    }

}

@RequiresApi(api = Build.VERSION_CODES.N)

@Override

public void onClick(View v) {

    if (daftarPenyakit.isEmpty()){

        showMessage(getString(R.string.msg_no_choice));

    }else{

        showLoading(getString(R.string.title_diagnosa),

            getString(R.string.please_wait));

        new Handler().postDelayed(this::diagnosis,2000);

    }

}

public void refresh(){

    G01.setChecked(false);

```

```
G02.setChecked(false);

G03.setChecked(false);

G04.setChecked(false);

G05.setChecked(false);

G06.setChecked(false);

G07.setChecked(false);

G08.setChecked(false);

G09.setChecked(false);

G10.setChecked(false);

G11.setChecked(false);

G12.setChecked(false);

G13.setChecked(false);

G14.setChecked(false);

G15.setChecked(false);

G16.setChecked(false);

G17.setChecked(false);

G18.setChecked(false);

}

@SuppressLint("DefaultLocale")

private void diagnosis(){
```



```

    rangkum.clear();

    rangkum.add(new penyakit("Rabun Jauh (Myopi)",
(checkPenyakitA(P01, daftarPenyakit))));

    rangkum.add(new penyakit("Rabun Dekat (Hipermetropi)",
checkPenyakitB(P02, daftarPenyakit))));

    rangkum.add(new penyakit("Katarak Senilis", checkPenyakitC(P03,
daftarPenyakit))));

    Comparator<penyakit> c = Collections.reverseOrder(new Sortbyroll());

    Collections.sort(rangkum, c);

    if (rangkum.get(0).getNilaiPresentase()==100 &&
rangkum.get(1).getNilaiPresentase()==100

        && rangkum.get(2).getNilaiPresentase()==100) {

        showMessage(ConsultActivity.this, String.format("%s%s%s
\n%s\n%s\n%s%d%%",

            name, String.format(" %s", getString(R.string.msgdiagnosa)) + " ",

            rangkum.get(0).getNamaPenyakit(),
rangkum.get(1).getNamaPenyakit(),

            rangkum.get(2).getNamaPenyakit(),
getString(R.string.msg_accuracy)+ " ",

            rangkum.get(0).getNilaiPresentase()));

        hideLoading();

    }else

        if (rangkum.get(0).getNilaiPresentase()==100 &&
rangkum.get(1).getNilaiPresentase()==100){

```

```

        showMessage(ConsultActivity.this,

            String.format("%s%s%s \n%s\n%s%d%%", name,
String.format(" %s",

                getString(R.string.msgdiagnosa))+ " ",

                rangkum.get(0).getNamaPenyakit(),
rangkum.get(1).getNamaPenyakit(),

                getString(R.string.msg_accuracy) + " ",
rangkum.get(0).getNilaiPresentase()));

        hideLoading();

    }else

        if (rangkum.get(0).getNilaiPresentase()==100 &&
rangkum.get(2).getNilaiPresentase()==100){

            showMessage(ConsultActivity.this, String.format("%s%s%s
\n%s\n%s%d%%",

                name, String.format(" %s", getString(R.string.msgdiagnosa))+ " ",

                rangkum.get(0).getNamaPenyakit(),

                rangkum.get(2).getNamaPenyakit(),
getString(R.string.msg_accuracy) + " ",

                rangkum.get(0).getNilaiPresentase()));

            hideLoading();

        }else

            if (rangkum.get(1).getNilaiPresentase()==100 &&
rangkum.get(2).getNilaiPresentase()==100){

```

```

        showMessage(ConsultActivity.this, String.format("%s%s%s\n%s\n%s%d%%",

            name, String.format(" %s", getString(R.string.msgdiagnosa))+ " ",

            rangkum.get(1).getNamaPenyakit(),
            rangkum.get(2).getNamaPenyakit(),

            getString(R.string.msg_accuracy)+ " ",
            rangkum.get(0).getNilaiPresentase()));

        hideLoading();

    } else {

        showMessage(ConsultActivity.this, String.format("%s%s%s\n%s%d%%", name,

            String.format(" %s", getString(R.string.msgdiagnosa)) + " ",

            rangkum.get(0).getNamaPenyakit(),
            getString(R.string.msg_accuracy)+ " ",

            rangkum.get(0).getNilaiPresentase()));

        hideLoading();

    }

    refresh();

}

private Integer checkPenyakitA(ArrayList<String> array1, ArrayList<String>
array2) {

    Iterator<String> itr = array2.iterator();

```

```

        A.clear();

        while (itr.hasNext()) {

            Object list_two_element = itr.next();

            if (array1.contains(list_two_element)) {

                A.add(list_two_element);

            }

        }

        int totaldipilih = A.size();

        double total = (Double.parseDouble(String.valueOf(totaldipilih)) / 12) *
100;

        return Integer.valueOf(String.valueOf(Math.round(total)));

    }

    private Integer checkPenyakitB(ArrayList<String> array1, ArrayList<String>
array2) {

        Iterator<String> itr = array2.iterator();

        A.clear();

        while (itr.hasNext()) {

            Object list_two_element = itr.next();

            if (array1.contains(list_two_element)) {

                A.add(list_two_element);

            }

        }

    }

```

```

    }

    int totaldipilih = A.size();

    double total = (Double.parseDouble(String.valueOf(totaldipilih)) / 9) * 100;

    return Integer.valueOf(String.valueOf(Math.round(total)));
}

```

```

private Integer checkPenyakitC(ArrayList<String> array1, ArrayList<String>
array2) {

    Iterator<String> itr = array2.iterator();

    A.clear();

    while (itr.hasNext()) {

        Object list_two_element = itr.next();

        if (array1.contains(list_two_element)) {

            A.add(list_two_element);

        }

    }

    int totaldipilih = A.size();

    double total = (Double.parseDouble(String.valueOf(totaldipilih)) / 8) * 100;

    return Integer.valueOf(String.valueOf(Math.round(total)));

}

}

```

Solution Activity

```
package com.hafiz.forwardchaining.activity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.ImageView;

import androidx.appcompat.app.AppCompatActivity;

import com.hafiz.forwardchaining.R;

public class SolutionActivity extends AppCompatActivity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_solution);

        ImageView btnBack = findViewById(R.id.btnBack);

        btnBack.setOnClickListener(v -> onBackPressed());

    }

    public void rabunjauh(View view) {

        Intent rabunJauh = new Intent(this,webView.class);

        rabunJauh.putExtra("namafile","rabunjauh.html");
```

```
        startActivity(rabunJauh);  
    }  
  
    public void rabundekat(View view) {  
  
        Intent rabunJauh = new Intent(this,webView.class);  
  
        rabunJauh.putExtra("namafile","rabundekat.html");  
  
        startActivity(rabunJauh);  
    }  
  
    public void katarak(View view) {  
  
        Intent rabunJauh = new Intent(this,webView.class);  
  
        rabunJauh.putExtra("namafile","katarak.html");  
  
        startActivity(rabunJauh);  
    }  
}
```

Lampiran 2

DAFTAR RIWAYAT HIDUP

Yang bertanda tangan dibawah ini :

Data Pribadi :

Nama Lengkap : Hafiz Al Irsyad
Tempat/Tanggal Lahir : Medan, 6 April 1997
Jenis Kelamin : Laki – Laki
Kewarganegaraan : Indonesia
Status Perkawinan : Belum Kawin
Agama : Islam
Alamat : Jalan Sibang Dusun I Senggani Gg. Aman B
Pendidikan Terakhir : Sarjana Ilmu Komputer
WA : 0813-1209-4131
Email : Hafizalirsyad0604@gmail.com



Pendidikan Formal :

1. SD Negeri 060826 (Lulus pada tahun 2009)
2. MTS Negeri 2 Medan (Lulus pada tahun 2012)
3. SMK Swasta Cerdas Murni (Lulus pada tahun 2015)
4. UIN Sumatera Utara Medan (Lulus pada tahun 2020)

Pengalaman Organisasi :

1. Ketua Bidang Pengabdian Masyarakat dan Perencanaan Kegiatan Organisasi Mahasiswa (Himpunan Mahasiswa Prodi Ilmu Komputer)
2. Ketua Bidang Minat dan Bakat (Dewan Eksekutif Mahasiswa Fakultas Sains Dan Teknologi).
3. Ketua Organisasi Club Kesenian Prodi Ilmu Komputer.
4. Manajer dan Pelatih di Club Basket Yayasan Islamic Centre Sumatera Utara.
5. Ketua Devisi Start-up di Industri Kreatif Medan.

Demikian curriculum vitae ini saya buat dengan sebenar-benarnya.

Medan, 13 Februari 2020
Hormat Saya

Hafiz Al Irsyad